

# New Solution Concepts, Algorithms, and Applications for Extensive-Form Games

*Learning, Correlation, Communication, and Common Knowledge*

Brian Hu Zhang

Proposal Date: August 23, 2024

Computer Science Department  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213

**Thesis Committee:**

Tuomas Sandholm (chair)  
Vincent Conitzer  
J. Zico Kolter  
Kevin Leyton-Brown (University of British Columbia)  
Roger B. Myerson (University of Chicago)

*Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy.*

DRAFT

This research was sponsored by the Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876; National Science Foundation grants RI-2312342, RI-1901403, IIS-1718457, and CCF-1733556; ARO awards W911NF2210266 and W911NF2010081; NIH award A240108S001; and a CMU Computer Science Department Hans Berliner PhD Student Fellowship.

## Abstract

Computational game theory has led to significant breakthroughs in AI dating back to the start of AI as a discipline. These include the strongest AI agents for both recreational and practical applications. For example, it has been instrumental in enabling superhuman AI from recreational games such as two-player zero-sum games chess, go, and heads-up poker to multiplayer games such as six-player poker and *Hanabi*, and even in games involving human language such as *Diplomacy*. It has also empowered a growing range of non-recreational applications, such as trading, machine learning robustness and safety, negotiation, conflict resolution, mechanism (*e.g.*, auction) design, information design, security, political campaigning, and self-driving cars.

This thesis pushes the boundary on computational game theory, especially in imperfect-information sequential (extensive-form) games, which are most prevalent in practical applications both in zero-sum games and beyond. We will present new **theoretical concepts and frameworks**, state-of-the-art and often provably optimal **algorithms for computing and learning equilibria**, and **new ways to apply such algorithms** to real-world problems, including problems in economics such as mechanism and information design.

The thesis contains four parts. Here, we highlight selected significant results from each part.

**Part I: Adversarial Team Games.** This part covers new solution concepts, algorithms, and complexity results for *adversarial team games*, which are games in which two teams compete against each other. We study two variants: one where each player’s team assignment is known, and one where some players’ team assignments may be hidden (“hidden-role games”). In the former case, we develop **optimal parameterized algorithms** where the parameter captures the amount of *asymmetric information* among team members. In the latter case, we develop the **first solution concept** suitable for hidden-role games and study its complexity. Under reasonable assumptions, we show that hidden-role games can be solved efficiently, and use our efficient algorithm to **exactly solve variants of the popular game *The Resistance: Avalon* with up to six players**.

**Part II: Generalized Mechanism Design.** This part covers applications of game solving to optimal generalized mechanism design. We develop a general framework that covers **sequential mechanism design, sequential information design, optimal correlated equilibria and more** for the first time, and **reduces them to zero-sum games**, thus enabling computation using any technique for computing equilibria in zero-sum games, **including (but not limited to) deep reinforcement learning**. For optimal correlation, we show that our framework is intrinsically connected to adversarial team games (as in the previous section), and this connection yields **the fastest algorithms for computing optimal correlated equilibria**.

**Part III: Learning in Games.** This part covers how agents can use *learning algorithms* to play games. The performance of a learning algorithm can be measured by the agent’s *regret*. Different notions of regret can be characterized by different sets of *strategy transformation functions* (“deviations”)—larger sets result in tighter notions of regret. We develop **the fastest learning algorithms** for minimizing regret against *linear* and *low-degree deviations*, which are the **tightest solution concepts known to be efficiently learnable in games**, based on a game-theoretic characterization of these deviation sets. We also show that a mediator with only the abilities to observe actions and pay the players can be used to **efficiently steer the players to any desired equilibrium** with necessary payments growing only sublinearly with the time horizon.

**Part IV: Subgame Solving.** This part covers new techniques for subgame solving in imperfect-information games. *Subgame solving* means performing computation *during a game* to choose an action, instead of playing from a precomputed strategy such as a policy network. It has been key to all the aforementioned major AI breakthroughs. We develop **new techniques that work even when the common-knowledge set is too large to work with** (which previous techniques cannot do), and use these techniques to build **the first strong bot**—and, to our knowledge, currently **the best bot—for the game of *dark chess***.

# Contents

<b>Introduction</b>	<b>7</b>
<b>1 Summary</b>	<b>7</b>
1.1 Part I: Equilibrium Computation in Adversarial Team Games . . . . .	7
1.2 Part II: Generalized Mechanism Design and Optimal Correlation via Zero-Sum Games . .	9
1.3 Part III: Learning in Games . . . . .	11
1.4 Part IV: Subgame Solving in Large Games . . . . .	12
1.5 Other Papers to be Added to the Dissertation . . . . .	12
<b>2 Preliminaries</b>	<b>13</b>
2.1 General Notation . . . . .	13
2.2 Extensive-Form Games . . . . .	13
2.2.1 Strategies . . . . .	14
2.2.2 Equilibria . . . . .	15
2.2.3 Tree-Form Decision Making . . . . .	15
2.3 No-Regret Learning and Counterfactual Regret Minimization . . . . .	16
2.3.1 Regret Minimization on Simplices . . . . .	16
2.3.2 Counterfactual Regret Minimization (CFR) . . . . .	18
2.3.3 Relation to Equilibrium Finding . . . . .	19
<b>I Equilibrium Computation in Adversarial Team Games</b>	<b>20</b>
<b>3 Efficient Parameterized Representations for Team and Imperfect-Recall Equilibrium Computation</b>	<b>20</b>
3.1 Introduction . . . . .	20
3.2 Preliminaries . . . . .	22
3.2.1 Behavioral and Mixed Max-Min Strategies . . . . .	22
3.2.2 Adversarial Team Games . . . . .	23
3.3 Beliefs and Observations . . . . .	25
3.3.1 Belief Game Construction . . . . .	29
3.3.2 Worst-Case Dimension of the Belief Game . . . . .	30
3.3.3 Regret Minimization on Team Games . . . . .	31
3.4 DAG Decision Problems . . . . .	32
3.5 DAG Decision Problems in Team Games . . . . .	34
3.5.1 Size Analysis of the TB-DAG . . . . .	35
3.5.2 Fixed-Parameter Hardness . . . . .	35
3.5.3 Branching Factor Reduction . . . . .	35
3.6 Complexity of Adversarial Team Games . . . . .	36
3.6.1 Behavioral Max-Min Strategies . . . . .	37
3.6.2 Mixed Nash Equilibria . . . . .	37
3.7 Discussion . . . . .	37
3.7.1 Public States vs Observations . . . . .	37
3.7.2 Tree vs DAG Representation . . . . .	39
3.7.3 Definition of Information Complexity and Comparison of Bounds . . . . .	39
3.7.4 Connection with Tree Decomposition . . . . .	40
3.7.5 Postprocessing Techniques that Can Be Used to Shrink the TB-DAG . . . . .	41
3.8 Experiments . . . . .	41
3.8.1 Experimental Setting . . . . .	42
3.8.2 Discussion of the Results . . . . .	43
3.9 Conclusion . . . . .	44

<b>4</b>	<b>Hidden-Role Games: Equilibrium Concepts and Computation</b>	<b>44</b>
4.1	Introduction	44
4.1.1	Main Modeling Contributions	46
4.1.2	Main Computational Contributions	48
4.1.3	Experiments: <i>Avalon</i>	50
4.1.4	Examples	50
4.2	Preliminaries	51
4.3	Equilibrium Concepts for Hidden-Role Games	52
4.3.1	Models of Communication	52
4.3.2	Split Personalities	53
4.3.3	Equilibrium Notions	54
4.4	Computing Hidden-Role Equilibria	54
4.4.1	Computing Private-Communication Equilibria	54
4.4.2	Computing No/Public-Communication Equilibria	56
4.5	Worked Example	57
4.6	Properties of Hidden-role Equilibria	58
4.6.1	The Price of Hidden Roles	58
4.6.2	Order of Commitment and Duality Gap	59
4.7	Experimental Evaluation: <i>Avalon</i>	59
4.8	Conclusions and Future Research	61
<b>II</b>	<b>Generalized Mechanism Design and Optimal Correlation via Zero-Sum Games</b>	<b>62</b>
<b>5</b>	<b>Polynomial-Time Optimal Equilibria with a Mediator</b>	<b>62</b>
5.1	Introduction	62
5.2	Preliminaries: Communication and Certification Equilibria	64
5.3	Extensive-Form $\mathcal{S}$ -Certification Equilibria	65
5.3.1	Proof of Theorem 5.3: The Single-Deviator Mediator-Augmented Game	65
5.3.2	Extensions and Special Cases	67
5.3.3	The Gap between Polynomial and Not Polynomial	68
5.3.4	A Family of Equilibria	69
<b>6</b>	<b>Optimal Correlated Equilibria in General-Sum Games: Fixed-Parameter Algorithms, Hardness, and Two-Sided Column-Generation</b>	<b>71</b>
6.1	Introduction	71
6.2	Preliminaries: Correlated Equilibria in Games	73
6.3	Example of Solution Concepts	75
6.4	Unifying Correlated Solution Concepts via Mediator-Augmented Games	76
6.4.1	Optimal Correlation via the Augmented Game	80
6.4.2	Comparison to Relevant Sequence-Based Construction of $\Xi$	81
6.5	Representing Imperfect-Recall Decision Spaces	82
6.5.1	Analyzing the Size of the Representation	83
6.5.2	Public Player Actions	83
6.6	Two-Player Games with Public Chance	84
6.6.1	Discussion: Relationship to Triangle-Freeness	85
6.6.2	Fixed-Parameter Hardness of Representing $\Xi^{\text{EFCCE}}$ and $\Xi^{\text{EFCE}}$	85
<b>7</b>	<b>Computing Optimal Equilibria and Mechanisms via Learning in Zero-Sum Games</b>	<b>86</b>
7.1	Introduction	86
7.2	Preliminaries	87
7.3	Lagrangian Relaxations and a Reduction to a Zero-Sum Game	88
7.3.1	“Direct” Lagrangian	88

7.3.2	Thresholding and Binary Search	90
<b>8</b>	<b>Experiments and Conclusion</b>	<b>91</b>
8.1	Optimal Equilibria in Tabular Games	91
8.2	Exact Sequential Auction Design	93
8.3	Scalable Sequential Auction Design via Deep Reinforcement Learning	94
8.4	Conclusion	94
<b>III</b>	<b>Learning in Games</b>	<b>96</b>
<b>9</b>	<b>Preliminaries</b>	<b>96</b>
9.1	$\Phi$ -Regret Minimization	96
9.2	The GGM Construction	97
9.3	Convergence to Correlated Equilibria	97
<b>10</b>	<b>Mediator Interpretation and Faster Learning Algorithms for Linear Correlated Equilibria</b>	<b>98</b>
10.1	Introduction	98
10.2	Mediators and UTC Deviations	99
10.3	Representation of UTC Deviations and Equivalence between UTC and Linear Deviations	100
10.4	Example	101
10.5	Regret Minimization on $\Phi_{\text{UTC}}$	101
10.6	Untimed Communication Equilibria	103
10.7	Experimental Evaluation	104
10.8	Conclusion	104
<b>11</b>	<b>Efficient <math>\Phi</math>-Regret Minimization with Low-Degree Swap Deviations</b>	<b>105</b>
11.1	Introduction	105
11.2	Our Results	106
11.3	Technical Overview	107
11.4	Hardness of Minimizing $\Phi$ -Regret in Behavioral Strategies	108
11.5	Circumventing Fixed Points	109
11.5.1	Approximate Expected Fixed Points	109
11.5.2	Extending Deviation Maps to $\text{co } \mathcal{X}$	110
11.5.3	Efficiently Computing Fixed Points in Expectation	111
11.5.4	Application for Faster Computation of Correlated Equilibria	112
11.6	Low-Degree Regret on the Hypercube	112
11.7	Extensive-Form Games	114
11.7.1	Interleaving Decision Problems	114
11.7.2	Efficient Low-Degree Swap-Regret Minimization in Extensive-Form Games	115
11.8	Discussion and Applications	118
11.8.1	Convergence to Correlated Equilibria	118
11.8.2	Strict Hierarchy of Equilibrium Concepts	118
11.8.3	Characterization of Recent Low-Swap-Regret Algorithms in Our Framework	119
11.8.4	Revelation Principles (or Lack Thereof)	119
11.9	Conclusions and Future Work	119
<b>12</b>	<b>Steering No-Regret Learners to a Desired Equilibrium</b>	<b>120</b>
12.1	Introduction	120
12.2	Summary of Our Results	120
12.3	The Steering Problem	123
12.4	Steering in Normal-Form Games	124
12.5	Steering in Extensive-Form Games	125
12.5.1	Steering with Full Feedback	125

12.5.2	Steering with Trajectory Feedback	126
12.5.3	Lower Bound	127
12.5.4	Upper Bound	127
12.6	Other Equilibrium Notions and Online Steering	128
12.6.1	Necessity of Advice	128
12.6.2	More General Equilibrium Notions: Bayes-Correlated Equilibrium	128
12.6.3	Online Steering	129
12.7	Experimental Results	131
12.8	Conclusions and Future Research	131
<b>IV Subgame Solving in Large Games</b>		<b>132</b>
<b>13</b>	<b>Subgame Solving without Common Knowledge</b>	<b>132</b>
13.1	Introduction	132
13.2	Preliminaries	133
13.3	Common-Knowledge Subgame Solving	133
13.4	Knowledge-Limited Subgame Solving	134
13.4.1	Safety by Updating the Blueprint	135
13.4.2	Safety by Allocating Deviations from the Blueprint	136
13.4.3	Affine Equilibrium, which Guarantees Safety against All Equilibrium Strategies	136
13.5	Example of How 1-KLSS Works	138
13.6	Dark Chess: An Agent from Only a Value Function Rather Than a Blueprint	138
13.7	Experiments	139
13.8	Conclusions and Future Research	140
<b>Future Research and Timeline</b>		<b>142</b>
<b>A</b>	<b>Toward a Superhuman Dark Chess Agent</b>	<b>142</b>
A.1	Learning-Based Game Solvers	142
A.1.1	The Node Expander	143
A.1.2	The Game Solver	144
A.2	Other Improvements	144
<b>B</b>	<b>Swap Regret and Complexity of NFCE</b>	<b>145</b>
B.1	Notation	146
B.2	Impossibility of Swap Regret Minimization	146
B.3	Polynomial-time expected fixed points	148
B.3.1	Ellipsoid against hope	149
B.3.2	Polynomial-time expected fixed points using ellipsoid against hope	149
B.4	Further Research	149
<b>C</b>	<b>Sequential Communication Equilibria</b>	<b>150</b>
C.1	Setup	150
C.2	Efficient Algorithm for SCE	152
C.3	Remaining Questions and Tasks	153
<b>D</b>	<b>Applications of Generalized Mechanism Design</b>	<b>153</b>
D.1	Setup	153
D.2	Characterization of Optimal Mechanisms	154
D.3	Example: Auctions with Budget Constraints	154
D.4	Proposed Research	155

# Introduction

## 1 Summary

Any intelligent agent—artificial or human—operating in an environment where there are other agents with their own incentives should be designed with game-theoretically sound methods, lest the agents perform suboptimally or even unsafely. This is a challenging task, but one that will be vital to the success of future intelligent agents.

To this end, computational game theory in sequential settings has led to numerous breakthroughs in AI dating back to the start of AI as a discipline. Perhaps most notably, it has led to the first superhuman-level AI agents for various games including classic two-player zero-sum games such as chess (*e.g.*, [Hsu 2002](#)), go ([Silver et al., 2016](#)), and heads-up poker ([Brown and Sandholm, 2018](#)); multiplayer games such as multiplayer poker ([Brown and Sandholm, 2019b](#)); identical-interest games such as *Hanabi* ([Lerer et al., 2020](#)); and even expert-level play in games involving human language such as *Diplomacy* ([Bakhtin et al., 2022](#)).

Despite these major advances, there remain many interesting problems to resolve in computational game theory, both in theory and in practice. This thesis aims to address some of these important outstanding problems.

This thesis is partitioned into four broad parts by general topic area. However, many of the results, even in different parts, are closely related to each other. We will point out relationships between the sections as they arise.

### 1.1 Part I: Equilibrium Computation in Adversarial Team Games

This part covers the following papers.

- Luca Carminati, Brian Hu Zhang, Federico Cacciamani, Junkang Li, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Efficient representations for team and imperfect-recall equilibrium computation. *in preparation*, 2024a (Subsumes [Zhang et al. \(2023b\)](#) and [Zhang and Sandholm \(2022b\)](#))
- Luca Carminati, Brian Hu Zhang, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Hidden-role games: Equilibrium concepts and computation. *ACM Conference on Economics and Computation (EC)*, 2024b

Outside two-player zero-sum games, notions of equilibrium run into the issue of *non-exchangeability*: if two players in a game independently compute (for example) Nash equilibria, their joint strategy may be arbitrarily bad unless they happen to have computed the same equilibrium, and there is no general way to pick a “best equilibrium”. This phenomenon limits the ability to apply natural equilibrium concepts beyond two-player zero-sum games. One of the only settings in which one *can* define natural solution concepts without running into exchangeability issues is the setting of *adversarial (zero-sum) team games*—that is, games in which there are two teams competing against each other, and their utilities are opposite (*e.g.*, one team wins and the other team loses). In such games, the key challenge is *asymmetric information* between different members of the same team—if all team members had the same information, we could simply treat the team as a single player with that common information.

We study two different variants of adversarial team games: one where the team assignment is common knowledge, and one where the team assignment may be *hidden* from one of the teams.

*Common-knowledge team assignments (Adversarial team games).* When the team assignment is common knowledge to all players, the game is simply called an *adversarial team game*. The most natural solution concept for adversarial team games is the *correlated team max-min equilibrium* (TMECor) (Basilico et al., 2017). TMECor arises as the mixed-strategy Nash equilibrium of the two-player zero-sum imperfect-recall game in which team members are merged into a single player. It represents the solution concept in which team members are allowed to discuss their strategy before the game (including flipping random coins which are not observed by the opposing team), but are not allowed to communicate once the game begins except as explicitly permitted by the game rules.

In Zhang and Sandholm (2022b) and Zhang et al. (2023b), we develop *parameterized algorithms* for computing TMECor in extensive-form adversarial team games. Our algorithm is based on the enumeration of *beliefs*. Intuitively, in this context, a belief is a minimal subset of nodes  $B$  such that there exists a strategy  $\mathbf{x}$  such that (1)  $\mathbf{x}$  reaches every node in  $B$ , and (2) upon a node in  $B$  being reached under  $\mathbf{x}$ , it is common knowledge among all team members that set  $B$  has been reached.

The time complexity analysis of our algorithm, roughly speaking, results from counting the number of beliefs. In particular, our algorithm scales at  $O^*((b+1)^k)$ , where

- $b$  is the branching factor,
- $k$  is the *information complexity*, a natural parameter that we define that characterizes in a sense the extent to which the information states of different members of the same team are *asymmetric*, and
- $O^*$  hides factors polynomial in the game size.

We show that these bounds are in a sense optimal: setting  $b = O(1)$  and  $k = O(n)$  can solve  $n$ -variable SAT, and the dependence on  $d$  for EFCCE and EFCE cannot be removed under ETH.

Our algorithm also enables the use of *regret minimization* for adversarial team games with the same time complexity. This is important in practice because regret minimizers are the fastest practical game solvers, and indeed we empirically show state-of-the-art performance across a wide variety of games using modern regret minimization techniques in combination with our construction.

Adversarial team games are equivalent to (timeable) two-player zero-sum games of imperfect recall. Thus, our results above can be thought of as a way of *representing the strategy space of an imperfect-recall player* with a size that is parameterized by the amount of asymmetric information.

Along the way, we also make two other contributions of independent interest in Zhang et al. (2023b).

- We classify precisely the complexity of TMECor and TME in adversarial team games. In particular, we show that computing the TME value<sup>1</sup> is  $\Sigma_2^P$ -complete, and computing the TMECor value is  $\Delta_2^P$ -complete<sup>2</sup>, thus exhibiting a strict separation between the two problems assuming that the polynomial hierarchy does not collapse.
- We define a notion of *DAG-form decision problem* that generalizes tree-form decision problems in a way that the strategy set is still a polytope and regret minimization is still possible. Our notion of DAG-form decision problem will be used multiple times throughout the remainder of this thesis in various settings that may at first seem unrelated.

---

<sup>1</sup>*i.e.*, deciding whether the value is at least some threshold  $t$ , up to exponentially-small error tolerance

<sup>2</sup>even with no error tolerance



**Hidden team assignments (Hidden-role games).** Recently (Carminati et al., 2024b), we were the first to conduct formal game-theoretic study of a class of games known as *hidden-role games*, where some players’ team allegiance is not common knowledge. Such games appear very frequently in both real-world and recreational applications. For example, consider a group of computers performing a task that requires information sharing among them. Some computers may have been corrupted by an adversary, but each computer may not know which other computers have been corrupted. Inadvertently sharing information with the adversary could lead to negative outcomes. How should the computers communicate and collaborate to achieve the best possible outcome? Hidden-role games also include well-known recreational games such as *Mafia* or *The Resistance*. Although hugely popular, this class of games has lacked formal study: it was previously not even clear how to define a solution concept. A reasonable solution concept should take into account that teams need to *collaborate*, and that *intra-team communication* can be compromised due to the presence of hidden roles. Our techniques therefore draw heavily from the literature on both *team games* (such as in the previous section) and cryptography (secure multi-party computation).

We developed a solution concept, which we call the *hidden-role equilibrium*, that satisfies both the above conditions. Further, we proved that hidden-role equilibria can—surprisingly—be found in polynomial time! We also showed bounds on the *price of hidden roles*, which we define as the factor by which a team would benefit if it knew the identities of all the adversaries. (This is analogous to the *price of anarchy* and *price of stability* that are common quantities to study in more traditional games.) From a practical standpoint, we used our techniques to exactly solve five- and six-player variants of the popular game *The Resistance: Avalon*.

## 1.2 Part II: Generalized Mechanism Design and Optimal Correlation via Zero-Sum Games

This part covers the following papers.

- Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. *ACM Conference on Economics and Computation (EC)*, 2022b
- Brian Hu Zhang and Tuomas Sandholm. Polynomial-time optimal equilibria with a mediator in extensive-form games. *arXiv preprint arXiv:2206.15395*, 2022a
- Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen McAleer, Andreas Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023a

We have developed a framework (Zhang and Sandholm, 2022a) that unifies a large family of game-theoretic problems under a single umbrella. An incomplete list of the problems that fall under the framework is the following.

- Computing an optimal (*e.g.*, social welfare-maximizing) *correlated equilibrium* in a general-sum game: specifically, optimal *extensive-form correlated equilibrium* (EFCE) (von Stengel and Forges, 2008), *extensive-form coarse correlated equilibrium* (EFCCE) (Farina et al., 2020), or *normal-form coarse correlated equilibrium* (NFCCE) (Moulin and Vial, 1978).<sup>3</sup>
- Computing an optimal *communication equilibrium* (Forges, 1986; Myerson, 1986). This problem includes—among others—the popular economic settings of *optimal sequential mechanism design* and Bayesian persuasion (information design) (Kamenica and Gentzkow, 2011) as special cases, and is sometimes referred to as *generalized mechanism design*.
- *Certification equilibria* (Forges and Koessler, 2005), which are communication equilibria in which certain messages are *verifiable*.

---

<sup>3</sup>In particular, I purposefully exclude the *normal-form correlated equilibrium* (Aumann, 1974), which is more difficult to reason about in extensive-form games and which we will discuss more in Part III.

The framework is based on the observation that all of these problems essentially boil down to *representing the strategy space for a certain “meta-agent”, or “mediator”*. (For example: for mechanism design, the mediator is the mechanism designer. For correlated equilibria, the mediator is the correlation device.) The main positive result of Zhang and Sandholm (2022a) is an efficient LP-based algorithm exists for computing an optimal equilibrium in this framework. For communication equilibrium, it runs in time polynomial in the size of the game tree. For certification equilibrium, it runs in polynomial time assuming a generalized form of the *nested range condition* (Green and Laffont, 1977). For correlated equilibria, it uses the TB-DAG algorithm described in Section 1.1 to represent the imperfect-recall decision space of the mediator.

Correlated equilibria are special in the above discussion in that the LP-based algorithm describe above is not necessarily efficient. In our framework, this is justified by the fact that the mediator for correlated equilibrium has *imperfect recall*, and (as discussed in Section 1.1) representing imperfect-recall decision spaces is hard in general. The relationship between imperfect-recall mediators and correlated equilibria gives rise to a different interpretation of correlated equilibria as *generalized mechanism design with privacy constraints*: in a sense, the imperfect recall of the mediator represents precisely the constraint that the mediator cannot *leak information between players*. Indeed, we can use this and other observations to write down an entire family of equilibria that include the three above bullets as special cases.

We also conducted a more in-depth study of correlated equilibrium notions specifically (Zhang et al., 2022b)<sup>4</sup>. In particular, the parameter of *information complexity* that we discussed in Section 1.1 can be generalized to also capture general games and optimal correlated equilibria. In this setting, we prove the bounds  $O^*((b+1)^k)$  for NFCCE,  $O^*((b+d)^k)$  for EFCCE, and  $O^*((bd)^k)$  for EFCE, where  $b$  and  $k$  are as in Section 1.1 and  $d$  is the game’s depth. Like the general imperfect-recall bounds in Section 1.1, we show that these bounds are in a sense optimal: setting  $b = O(1)$  and  $k = O(n)$  can solve NP-complete problems, and the dependence on  $d$  for EFCCE and EFCE cannot be removed under ETH.

In a recent paper (Zhang et al., 2023a), we developed techniques for the problems in this framework that allowed for the first time the application of deep reinforcement learning (RL) to this large family of problems, thus allowing for the possibility far greater scalability. Our techniques are based on reducing the general family of problems, via a Lagrangian relaxation, to a *zero-sum game*<sup>5</sup>. Thus, *if one can solve zero-sum games in extensive form, one can also compute solutions in the general framework described above, including optimal sequential generalized mechanisms*.

Our techniques here can be thought of as a generalization of the framework of *mechanism design with deep learning* first introduced by Dütting et al. (2019). Compared to that line of work, our techniques make two improvements. The first is, as above, generality: our techniques work for arbitrary communication equilibria (“generalized mechanisms”) and in sequential settings. The second is an *improved Lagrangian formulation* that does not depend on a Lagrange multiplier that needs to either be known *a priori* or grow arbitrarily large. This results in a method that is significantly easier to work with, especially in a deep learning setting where a large Lagrange multiplier would correspond to the need for deep RL to achieve extremely precise results. In experiments, we show that the learning-based algorithms are faster than the LP-based algorithms of Zhang and Sandholm (2022a), and that the improved Lagrangian formulation is critical to performance with deep RL.

---

<sup>4</sup>The three papers (1) Zhang et al. (2023b), (2), Zhang et al. (2022b), and (3) Zhang and Sandholm (2022a) were written and appeared as preprints in that order (1, 2, 3). However, they appeared in conference publication in the order 2, 3, 1, and in this thesis we discuss them in the order 1, 3, 2, because that is the cleanest conceptual introduction despite corresponding to neither the preprint appearance order nor the publication order.

<sup>5</sup>as before, with imperfect recall in the case of correlated equilibria

### 1.3 Part III: Learning in Games

This part covers the following papers.

- Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Mediator interpretation and faster learning algorithms for linear correlated equilibria in general sequential games. *International Conference on Learning Representations (ICLR)*, 2024d
- Brian Hu Zhang, Ioannis Anagnostides, Gabriele Farina, and Tuomas Sandholm. Efficient  $\Phi$ -regret minimization with low-degree swap deviations in extensive-form games. *arXiv preprint arXiv:2402.09670*, 2024a
- Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen Marcus McAleer, Andreas Alexander Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Steering no-regret learners to optimal equilibria. *ACM Conference on Economics and Computation (EC)*, 2024b

One frontier of research in computational game theory studies how *uncoupled learning agents* behave when interacting with an environment (for example, a game) over time. The performance of a learning algorithm can be measured by its *regret*—that is, the improvement in reward that the agent would have experienced had it played other strategies instead. Different notions of regret therefore can be characterized by different sets of functions mapping the strategy played by the agent to other strategies that could have been more profitable—accommodating larger sets of functions results in more robust notions of regret.

**Linear and Low-Degree Swap Regret.** We have developed (Zhang et al., 2024d) the fastest algorithm for minimizing *linear-swap regret* (Farina and Pipis, 2023), which is the regret against the set of all *linear* functions. It takes time polynomial in the number of nodes  $N$  and precision  $\epsilon$ . The algorithm is based on a perhaps-surprising relationship between the set of linear functions and the set of *untimed communication deviations*, which intuitively resemble the set of deviations in a communication equilibrium (as in the previous section), except that players are not constrained to send a single message at every timestep.

In a recent preprint (Zhang et al., 2024a), we extend this analysis to *low-degree polynomials*. In particular, we show that that low regret against the set of *degree- $k$  polynomials* can be minimized in time  $N^{O(kd)^3}/\epsilon^2$  if the game has size  $N$  and depth  $d$ . This result smoothly interpolates between linear-swap regret ( $k = 1$ ) and swap regret ( $k = N$ ), almost matching the aforementioned bounds at either extreme. Our algorithm works by extending the framework of Gordon et al. (2008) to nonlinear deviations by using *expected fixed points*<sup>6</sup>, and then extending the relationship between linear-swap deviations and untimed communication deviations to also encompass low-degree polynomials, by using multiple mediators.

**Steering No-Regret Learners.** We have shown in a recent preprint (Zhang et al., 2024b) that if we allow an external observer (a mediator) to help *steer* the players, much stronger guarantees, such as convergence to *Nash equilibrium*, can be achieved. We introduced a mediator with the power to provide *payments* to the players, and we showed that the ability of the mediator to succeed in steering depends on how much budget the mediator has as a function of the time: if the mediator’s budget is constant, we showed that no steering is possible; if the mediator’s budget grows linearly with time, the mediator can trivially steer the players toward any behavior by simply providing large enough payments. The case of *sublinearly* growing payments is therefore the most interesting case, and indeed we showed that, with reasonable assumptions, a mediator can steer any no-regret players toward any equilibrium of its choice with only a total budget that increases *sublinearly* with the time horizon.

---

<sup>6</sup>An *expected fixed point* of a function  $\phi : \mathcal{X} \rightarrow \mathcal{X}$ , where  $\mathcal{X}$  is convex and bounded, is a distribution  $D \in \Delta(\mathcal{X})$  such that  $\mathbb{E}_{\mathbf{x} \sim D} \|\phi(\mathbf{x}) - \mathbf{x}\| = 0$ . The critical property here is that approximate expected fixed points are significantly easier to compute than actual fixed points (*i.e.*, points  $\mathbf{x}$  for which  $\phi(\mathbf{x}) = \mathbf{x}$ ).

## 1.4 Part IV: Subgame Solving in Large Games

This part covers the following paper.

- Brian Hu Zhang and Tuomas Sandholm. Subgame solving without common knowledge. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021b

Subgame solving is the idea that one should *refine* a strategy online while playing the game, instead of playing solely from some precomputed strategy such as a policy network. As an idea, it is perhaps older than AI as a field<sup>7</sup> and has been vital in all of the breakthroughs mentioned in the first paragraph. In perfect-information settings, it has been used since the beginnings of AI as a field and has been fundamental to the success of strong agents—for example, the superhuman chess agent *Leela Chess Zero* drops to “only” human expert level without subgame solving, but is easily superhuman with subgame solving. However, the application of subgame solving to imperfect-information settings, especially in a game-theoretically safe manner, is much more challenging, and has only been studied recently (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017). These techniques were one of the core ingredients of the superhuman breakthroughs in no-limit Texas hold’em (NLTH) poker (Brown and Sandholm, 2018, 2019b).

All prior techniques for safe subgame solving suffer from a shared weakness that limits their applicability: they require reasoning about the *common-knowledge closure* of the player’s current information set—that is, the smallest set of states in which it is common knowledge that the current state lies. In poker, this set is manageable; however, in many other games, it is not. I developed *knowledge-limited subgame solving* (KLSS) (Zhang and Sandholm, 2021b), which is the first known technique that does not have this weakness. Instead, this technique can work by only expanding the nodes that are still reachable in the game tree from the player’s current information set. We use our technique to implement, to our knowledge, the first and currently strongest agent for the game *dark chess*<sup>8</sup>, an imperfect-information variant of chess in which common-knowledge closures are too large to be tackled by prior subgame-solving techniques.

This research is a new way of thinking about subgame solving with imperfect information, and has already led to impact. We specifically emphasize the work of Liu et al. (2023), who extended our methods to develop a more theoretically sound version of KLSS, and applied it to achieve improved performance in Mahjong.

## 1.5 Other Papers to be Added to the Dissertation

In the interest of not making the proposal longer than it already is, multiple already-accepted or published papers are excluded from this proposal but will be included in the thesis. In reverse chronological order, these are:

- Brian Hu Zhang and Tuomas Sandholm. Exponential lower bounds on the double oracle algorithm in zero-sum games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024a
- Brian Hu Zhang and Tuomas Sandholm. On the outcome equivalence of extensive-form and behavioral correlated equilibria. *AAAI Conference on Artificial Intelligence (AAAI)*, 2024b
- Brian Hu Zhang, Luca Carminati, Federico Cacciamani, Gabriele Farina, Pierricardo Olivieri, Nicola Gatti, and Tuomas Sandholm. Subgame solving in adversarial team games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022a
- Brian Hu Zhang and Tuomas Sandholm. Finding and certifying (near-)optimal strategies in black-box extensive-form games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021a
- Brian Hu Zhang and Tuomas Sandholm. Small Nash equilibrium certificates in very large games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020a
- Brian Hu Zhang and Tuomas Sandholm. Sparsified linear programming for zero-sum equilibrium finding. *International Conference on Machine Learning (ICML)*, 2020b

---

<sup>7</sup>For example, Alan Turing and David Champernowne wrote a chess engine *Turochamp* in 1948 using minimax search and node heuristics, which can be considered a form of subgame solving.

<sup>8</sup>also known as “fog of war chess” on chess.com, the most popular website on which this variant is played among humans

The following paper *may* also be included in part:

- Emanuel Tewelde, Brian Hu Zhang, Caspar Oesterheld, Manolis Zampetakis, Tuomas Sandholm, Paul Goldberg, and Vincent Conitzer. Imperfect-recall games: Equilibrium concepts and their complexity. *International Joint Conference on Artificial Intelligence (IJCAI), 2024*

## 2 Preliminaries

Here, we introduce the various pieces of background information that will be repeatedly referenced throughout this thesis. Background information that is more specialized to a single section of the thesis is deferred to that section.

### 2.1 General Notation

Unless otherwise stated, we will use the following notation:

- Vectors  $\mathbf{x} \in \mathbb{R}^n$  will be in italic boldface, as will generic indices into such vectors, which will be denoted either  $\mathbf{x}[i]$  or  $\mathbf{x}_i$ . Matrices will be in non-italic boldface, *e.g.*,  $\mathbf{A}$ .
- If  $A, B$  are sets then  $A^B$  is the set of functions  $f : B \rightarrow A$ .
- $f \lesssim g$  means  $f = O(g)$ . Similarly,  $f \gtrsim g$  means  $f = \Omega(g)$ , and  $f \sim g$  means  $f = \Theta(g)$ .
- $\circ$  denotes element-wise multiplication of vectors or matrices.
- $\Delta(S)$  is the probability simplex on set  $S$ , that is,  $\Delta(S) := \{\mathbf{x} \in \mathbb{R}_{\geq 0}^S : \sum_{s \in S} \mathbf{x}(s) = 1\}$ . If  $\mathbf{x} \in \Delta(S)$  then  $\text{supp } \mathbf{x}$  denotes the support of  $\mathbf{x}$ .
- $\text{co } S$  denotes the convex hull of a set  $S$ .
- $\tilde{O}, \tilde{\Omega}, \tilde{\Theta}$  hide logarithmic factors. That is,  $f = \tilde{O}(g)$  if  $f = O(g \log^k g)$ ;  $f = \tilde{\Omega}(g)$  if  $f = \Omega(g \log^{-k} g)$  (where in both cases  $k$  is an absolute constant), and  $f = \tilde{\Theta}(g)$  if  $f = \tilde{O}(g)$  and  $f = \tilde{\Omega}(g)$ .
- For any set  $S$ ,  $\text{Id} : S \rightarrow S$  is the identity function.
- $\mathbb{1}\{b\}$  is the indicator of the condition  $b$
- $[n] = \{1, \dots, n\}$ .
- $[x]^+ = \max(x, 0)$ .

### 2.2 Extensive-Form Games

*Extensive-form games* are the focus of the majority of this thesis. A finite extensive-form game (hereafter simply *game*)  $\Gamma$  with  $n$  players consists of the following components.

1. A tree of *nodes* or *histories*  $\mathcal{H}$ , rooted at a root history  $\emptyset \in \mathcal{H}$ . The leaf nodes of  $\mathcal{H}$  are called *terminal nodes*, and  $\mathcal{Z}$  will denote the set of terminal nodes. The edges out of a given node  $h$  are identified with *actions*  $a \in \mathcal{A}$ , and the subset of actions legal at  $h$  is  $\mathcal{A}(h)$ . The child of  $h$  reached by following action  $a$  is denoted  $ha$ . The *branching factor*  $b = \max_n |\mathcal{A}(h)|$  is the maximum number of legal moves at any node.
2. A partition  $\mathcal{H} \setminus \mathcal{Z} = \mathcal{H}_C \sqcup \mathcal{H}_1 \sqcup \dots \sqcup \mathcal{H}_n$ , where  $\mathcal{H}_i$  is the set of nodes at which player  $i$  acts, and player 0 is chance (or *nature*).
3. For each player  $i \in [n]$ , a *utility function*  $u_i : \mathcal{Z} \rightarrow [-1, 1]$  giving player  $i$ 's utility for reaching any given terminal node.
4. For each player  $i \in [n]$ , a partition  $\mathcal{I}_i$  of  $\mathcal{H}_i$  into *information sets*, or *infosets*. Any two nodes  $h, h'$  in the same infoset  $I \in \mathcal{I}_i$  must have the same set of legal actions, which we will denote  $\mathcal{A}(I)$ .
5. For each node  $h \in \mathcal{H}_C$ , a probability distribution  $x_C(\cdot|h) \in \Delta(\mathcal{A}(h))$ , denoting how chance selects its action at node  $h$ .

The game tree induces a natural ordering  $\preceq$  on sets of nodes: we will write  $S \preceq S'$  if there are histories  $h \in S, h' \in S'$  such that  $h'$  is a descendant of  $h$ . If either  $S$  or  $S'$  is a singleton, we will omit the braces: for

example,  $h \preceq h'$  denotes that  $h'$  is a descendant of  $h$ . We will use  $|h|$  to denote the depth of history  $h$ : that is,  $|\emptyset| = 0$  and  $|ha| = |h| + 1$ . The depth of game  $\Gamma$  is the maximum depth of any history.

**Perfect recall.** At a history  $h \in \mathcal{H}$ , the *sequence*  $\sigma_i(h)$  of player  $i$  is the list of information sets  $I \in \mathcal{I}_i$  encountered by player  $i$  on the path to  $h$ , and actions taken at those information sets, not including at  $h$  itself. We say that a player  $i$  has *perfect recall* if, for every info set  $I$ , every history  $h \in I$  has the same sequence, which we will denote  $\sigma_i(I)$  and call the *parent sequence* of  $I$ . The game  $\Gamma$  has perfect recall if all of its players do. We will denote by  $\Sigma_i$  the set of sequences of a player  $i$ .

**Timeability.** An extensive-form game is *timeable* if any path from the root to any node in the same info set has the same length (*i.e.* all histories belonging to the same info set have the same depth). Formally, the game is timeable if for every info set  $I \in \mathcal{I}$  and every  $h, h' \in I$ , we have  $|h| = |h'|$ .

## 2.2.1 Strategies

A *pure strategy* of player  $i$  is a choice of one action per info set of player  $i$ . The *realization form* of a pure strategy is the vector  $\mathbf{x}_i \in \{0, 1\}^{\mathcal{Z}}$  where  $\mathbf{x}_i[z] = 1$  if and only if the player plays all the actions on the  $\emptyset \rightarrow z$  path.

A *mixed strategy* is a distribution  $\pi_i \in \Delta(\mathcal{X}_i)$ . In many cases, we will only care about the realization form of a mixed strategy, which is simply defined to be  $\mathbb{E}_{\mathbf{x}_i \sim \pi_i} \mathbf{x}_i$ . The set of realization-form mixed strategies is hence  $\text{co } \mathcal{X}_i$ . A mixed strategy is *behavioral* if its action choices at different information sets are independent.

Multiple strategies can have the same realization form. If so, we will call those strategies (*realization-equivalent*). Unless otherwise stated, we will not distinguish between realization-equivalent strategies. *Kuhn's theorem* (Kuhn, 1953) guarantees that, for players with perfect recall, every mixed strategy is equivalent to a behavioral strategy, and thus it is usually without loss of generality to work with behavioral strategies (although we will see in Section 10 that it is not *always* the case!)

A *correlated strategy profile* (or simply *correlated profile*) is a distribution  $\pi \in \Delta(\mathcal{X}_1 \times \dots \times \mathcal{X}_n)$ . If  $\pi$  factors as a product distribution  $\pi = (\pi_1, \dots, \pi_n) \in \Delta(\mathcal{X}_1) \times \dots \times \Delta(\mathcal{X}_n)$ , we will drop the word *correlated* and simply call  $\pi$  a *strategy profile* or *profile*. If the word *correlated* is not used, all profiles are assumed to be uncorrelated. For uncorrelated profiles, we will usually circumvent writing the distribution at all, by expressing each player's mixed strategy  $\pi_i$  as a realization-form mixed strategy and thus expressing  $\pi$  as a tuple  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \text{co } \mathcal{X}_1 \times \dots \times \text{co } \mathcal{X}_n$ .

Every profile induces a distribution over terminal nodes, that results from sampling a pure profile  $\mathbf{x} \sim \pi$  and following those actions through the game, sampling chance actions where needed. We will use  $z \sim \pi$  (or  $z \sim \mathbf{x}$ ) to denote a sample from this distribution. The *expected value* of player  $i$  under profile  $\pi$ , denoted  $u_i(\pi)$ , is defined in the natural manner:

$$u_i(\pi) := \mathbb{E}_{z \sim \pi} u_i(z).$$

We will sometimes use *partial profiles*, which are profiles defined for only a subset of players. In particular, if  $\pi$  is a (possibly correlated) profile then we will use  $\pi_{-i}$  to denote its marginal on all players except  $i$ .

For uncorrelated profiles  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ , it is critical to note that the expected value  $u_i(\mathbf{x})$  is linear in each player's strategy. That is, for every fixed opponent profile  $\mathbf{x}_{-j}$ , the expected value  $u_i(\mathbf{x}_j, \mathbf{x}_{-j})$  is linear in  $\mathbf{x}_j$ . In particular, we have

$$u_i(\mathbf{x}) = \sum_{z \in \mathcal{Z}} x_C[z] u(z) \prod_{i=1}^n \mathbf{x}_i[z]$$

where  $x_C[z]$  is the probability that chance plays all actions on the path to  $z$ .

A game is *two-player zero-sum* if there are two players (which will always be denoted  $\blacktriangle$  and  $\blacktriangledown$ ), and  $u_{\blacktriangle} = -u_{\blacktriangledown}$ . In this case, we will generally use the notation  $u := u_{\blacktriangle}$ ,  $\mathcal{X} = \mathcal{X}_{\blacktriangle}$ , and  $\mathcal{Y} = \mathcal{X}_{\blacktriangledown}$ .

## 2.2.2 Equilibria

For our purposes, to *solve* a game will mean to find an *equilibrium* of it, for some *equilibrium concept* of interest. Here we identify some equilibrium concepts that we will use throughout the paper.

The *Nash equilibrium* (Nash, 1950) is the oldest and best-known notion of equilibrium for general games. An  $\epsilon$ -Nash equilibrium is an uncorrelated strategy profile  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \text{co } \mathcal{X}_1 \times \dots \times \text{co } \mathcal{X}_n$  such that no player can improve by more than  $\epsilon$  using any unilateral deviation:

$$u_i(\mathbf{x}'_i, \mathbf{x}_{-i}) \leq u_i(\mathbf{x}_i, \mathbf{x}_{-i}) + \epsilon$$

for every  $i \in [n]$  and  $\mathbf{x}'_i \in \mathcal{X}_i$ . Every game has a Nash equilibrium in mixed strategies.

Throughout this thesis, in various places we will also be interested in various notions of *correlated equilibria*. In the greatest possible generality, a notion of correlated equilibrium is defined by a tuple of sets of transformations  $\Phi = (\Phi_1, \dots, \Phi_n)$ , where  $\Phi_i \subseteq (\text{co } \mathcal{X}_i)^{\mathcal{X}_i}$  is a set of transformations of player  $i$ 's strategies. Then an  $\epsilon$ - $\Phi$ -equilibrium is a *correlated* profile for which

$$\mathbb{E}_{\mathbf{x} \sim \pi} [u_i(\phi_i(\mathbf{x}_i), \mathbf{x}_{-i}) - u_i(\mathbf{x}_i, \mathbf{x}_{-i})] \leq \epsilon$$

for every  $i \in [n]$  and  $\mathbf{x}'_i \in \mathcal{X}_i$ . Two extremes of this definition are the *normal-form coarse-correlated equilibrium* (NFCCE), for which  $\Phi_i$  is the set of all constant transformations  $\{\phi_{\mathbf{x}_i^*} : \mathbf{x}_i \mapsto \mathbf{x}_i^* \mid \mathbf{x}_i^* \in \mathcal{X}_i\}$ , and the *normal-form correlated equilibrium* (NFCE), for which  $\Phi_i = (\text{co } \mathcal{X}_i)^{\mathcal{X}_i}$  is the set of all possible transformations.

In zero-sum games, all the notions of correlated equilibria collapse to Nash equilibria<sup>9</sup>, and the Nash equilibria are precisely the saddle-point solutions  $(\mathbf{x}, \mathbf{y})$  to the convex bilinear saddle-point problem

$$\max_{\mathbf{x} \in \text{co } \mathcal{X}} \min_{\mathbf{y} \in \text{co } \mathcal{Y}} u(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{x} \in \text{co } \mathcal{X}} \min_{\mathbf{y} \in \text{co } \mathcal{Y}} \sum_{z \in \mathcal{Z}} p(z) u(z) \mathbf{x}(z) \mathbf{y}(z) = \max_{\mathbf{x} \in \text{co } \mathcal{X}} \min_{\mathbf{y} \in \text{co } \mathcal{Y}} \mathbf{x}^\top \mathbf{A} \mathbf{y} \quad (1)$$

where  $p(z)$  is the probability that chance plays all actions on the path to  $z$ , and the matrix  $\mathbf{A}$  is defined by  $\mathbf{A}[i, j] = \sum_{z \in \mathcal{Z}: \sigma_\blacktriangle(z)=i, \sigma_\blacktriangledown(z)=j} p(z) u(z)$ . We will call the saddle-point value of (1) the *equilibrium value* of  $\Gamma$ , and denote it  $u^*$ . Nash equilibria in zero-sum games are hence *exchangeable*: if  $(\mathbf{x}^1, \mathbf{y}^1)$  and  $(\mathbf{x}^2, \mathbf{y}^2)$  are Nash equilibria, then so are  $(\mathbf{x}^1, \mathbf{y}^2)$  and  $(\mathbf{x}^2, \mathbf{y}^1)$ .

## 2.2.3 Tree-Form Decision Making

It will be convenient at various points in the paper to abstract away the majority of a game and focus solely on the decision problem faced by a single player. When this happens, we will generally omit the subscript  $i$ ; for example,  $\mathbf{x}$  will denote a generic strategy for the player. For a perfect-recall player, this decision problem can be described as a *tree-form decision problem*. A tree-form decision problem consists of a tree of nodes  $\mathcal{T}$ , that are each one of two types:

- *decision points*  $j \in \mathcal{J}$ , at which the player must select an action  $a \in A(j)$ , and
- *observation points*  $\sigma \in \Sigma$ , at which the player makes an observation.

For a perfect-recall player in an extensive-form game, the decision and observation points correspond respectively to the information sets and sequences of that player. Unless otherwise stated, we will assume that decision and observation points alternate, and that the root  $\emptyset$  is an observation point—both of these are without loss of generality. The observation point child of  $j$  reached by taking action  $a$  is denoted  $ja$ , and the parent of  $j$  is denoted  $p_j$ . The set of children of  $\sigma$  is denoted  $C_\sigma$ . For notational simplicity, when  $\mathbf{x} \in \mathbb{R}^\Sigma$  is any vector indexed by observation points and  $j$  is a decision point, we will use  $\mathbf{x}[j^*] \in \mathbb{R}^{A(j)}$  to denote the subvector of  $\mathbf{x}$  indexed only by the children of  $j$ .

We now define strategies in tree-form decision problems analogously to strategies in games. A *pure strategy* is a choice of one action at every decision point. The *sequence form* of a pure strategy is the vector  $\mathbf{x} \in \mathcal{X}$

<sup>9</sup>In particular, one can show that, for any  $\epsilon$ -NFCCE, the product distribution with the same marginals is a  $2\epsilon$ -Nash equilibrium

---

**Algorithm MWU:** Multiplicative weight update  $\Delta(n)$ .

---

- 1: **initialize**  $\mathbf{z}^1 \leftarrow \mathbf{1}, t \leftarrow 0$
  - 2: **procedure** NEXTSTRATEGY(): **return**  $\mathbf{x}^t := \mathbf{z}^t / \|\mathbf{z}^t\|_1$
  - 3: **procedure** OBSERVEUTILITY( $\mathbf{u}^t$ ):  $\mathbf{z}^{t+1} \leftarrow \mathbf{z}^t \circ \exp(\eta \mathbf{u}^t)$
- 

**Algorithm RM+:** Regret matching plus on  $\Delta(n)$ .

---

- 1: **initialize**  $\mathbf{z}^1 \leftarrow \mathbf{0}, t \leftarrow 0$
  - 2: **procedure** NEXTSTRATEGY():
  - 3:     **if**  $\mathbf{z}^t = \mathbf{0}$  **then return**  $\mathbf{x}^t := \mathbf{z}^t / \|\mathbf{z}^t\|_1$
  - 4:     **else return**  $\mathbf{x}^t := \text{any strategy}$
  - 5: **procedure** OBSERVEUTILITY( $\mathbf{u}^t$ ):  $\mathbf{z}^{t+1} \leftarrow [\mathbf{z}^t + \mathbf{u}^t - \langle \mathbf{u}^t, \mathbf{x}^t \rangle]^+$
- 

indexed by sequences  $\sigma \in \Sigma$ , for which  $\mathbf{x}_i[\sigma] = 1$  if and only if the player plays all actions on the  $\emptyset \rightarrow \sigma$  path in  $\mathcal{T}$ . The sequence-form mixed strategies are then, once again, the convex hull of  $\mathcal{X}$ . Conveniently, the sequence-form mixed strategies are precisely the strategies obeying a natural family of linear constraints (von Stengel, 1996; Romanovskii, 1962):

$$\text{co } \mathcal{X} = \left\{ \mathbf{x} \in \mathbb{R}_{\geq 0}^{\mathcal{S}} \mid \mathbf{x}[\emptyset] = 1, \quad \mathbf{x}[p_j] = \sum_{a \in A(j)} \mathbf{x}[ja] \quad \forall j \in \Sigma \right\}.$$

Clearly, the sequence-form and realization-form representations are equivalent: given a sequence-form vector  $\mathbf{x}_i$  for a player  $i$ , one recovers the realization form by  $\mathbf{x}_i[z] := \mathbf{x}_i[\sigma_i(z)]$ . Which we choose to use will depend on which is most convenient. In both cases we will denote the set of pure strategies by  $\mathcal{X}_i$ .

## 2.3 No-Regret Learning and Counterfactual Regret Minimization

*No-regret learning* is a popular framework for decision making in repeated settings. As we will see, algorithms based on no-regret learning are the most popular and fastest practical algorithms for equilibrium computation. In this section we will discuss only algorithms for *external* regret minimization in extensive-form games; we defer the extension to the more general notion of  $\Phi$ -regret to Part III.

A decision maker is faced with the following interaction with an adversary. There is a convex *strategy set*  $\mathcal{X}$ , which for our purposes will always be a subset of  $[0, 1]^n$  and usually be the set of mixed sequence-form strategies of some tree-form decision problem. The interaction lasts for  $T$  timesteps. At each timestep  $t$ , the decision maker selects a point  $\mathbf{x}^t \in \text{co } \mathcal{X}$ . The adversary, observing  $\mathbf{x}^t$ , selects a utility vector  $\mathbf{u}^t \in \mathbb{R}^n$  such that  $\langle \mathbf{u}^t, \mathbf{x} \rangle \in [-1, 1]$  for all  $\mathbf{x} \in \mathcal{X}$ . After  $T$  timesteps, the (*averaged, external*) regret is defined as

$$\text{REG}(T) := \max_{\mathbf{x} \in \mathcal{X}} \frac{1}{T} \sum_{t=1}^T \langle \mathbf{u}^t, \mathbf{x} - \mathbf{x}^t \rangle.$$

### 2.3.1 Regret Minimization on Simplices

The most basic setting for no-regret learning is the setting in which  $\mathcal{X}$  is the simplex  $\Delta(n)$ . Here, we introduce two simple no-regret learning algorithms on the simplex. Here, we review two common regret minimization algorithms which we will refer to repeatedly throughout this thesis, and some important variants of them.



---

**Algorithm OMWU:** Predictive (optimistic) multiplicative weight update  $\Delta(n)$ .

---

- 1: **initialize**  $\mathbf{z}^1 \leftarrow \mathbf{1}$ ,  $t \leftarrow 0$
  - 2: **procedure** NEXTSTRATEGY( $\tilde{\mathbf{u}}^t$ ):
  - 3:      $\tilde{\mathbf{z}}^t \leftarrow \mathbf{z}^t \circ \exp(\eta \tilde{\mathbf{u}}^t)$
  - 4:     **return**  $\mathbf{x}^t := \tilde{\mathbf{z}}^t / \|\tilde{\mathbf{z}}^t\|_1$
  - 5: **procedure** OBSERVEUTILITY( $\mathbf{u}^t$ ):  $\mathbf{z}^{t+1} \leftarrow \mathbf{z}^t \circ \exp(\eta \mathbf{u}^t)$
- 

**Algorithm PRM+:** Predictive (optimistic) regret matching plus on  $\Delta(n)$ .

---

- 1: **initialize**  $\mathbf{z}^1 \leftarrow \mathbf{0}$ ,  $t \leftarrow 0$
  - 2: **procedure** NEXTSTRATEGY( $\tilde{\mathbf{u}}^t$ ):
  - 3:      $\tilde{\mathbf{z}}^t \leftarrow [\mathbf{z}^t + \tilde{\mathbf{u}}^t - \langle \tilde{\mathbf{u}}^t, \mathbf{x}^{t-1} \rangle]^+$
  - 4:     **if**  $\tilde{\mathbf{z}}^t = \mathbf{0}$  **then return**  $\mathbf{x}^t := \tilde{\mathbf{z}}^t / \|\tilde{\mathbf{z}}^t\|_1$
  - 5:     **else return**  $\mathbf{x}^t := \text{any strategy}$
  - 6: **procedure** OBSERVEUTILITY( $\mathbf{u}^t$ ):  $\mathbf{z}^{t+1} \leftarrow [\mathbf{z}^t + \mathbf{u}^t - \langle \mathbf{u}^t, \mathbf{x}^t \rangle]^+$
- 

**Multiplicative Weight Update.** The *multiplicative weights* algorithm is given in Algorithm MWU. It is parameterized by a single hyperparameter  $\eta > 0$ , called the *step size*. Multiplicative weights satisfies the following regret bound.

**Proposition 2.1.** *The average external regret of MWU satisfies:*

$$\text{REG}_{\text{MWU}}(T) \lesssim \frac{\log n}{\eta T} + \eta \lesssim \sqrt{\frac{\log n}{T}}$$

where the equality follows by taking the step size  $\eta = \sqrt{(\log n)/T}$ .

**Regret Matching Plus.** The *regret matching* algorithm (Hart and Mas-Colell, 2000) is a simple, hyperparameter-free no-regret learning algorithm. Here, we will introduce a better, more recent variants of it, known as *regret matching plus* (Algorithm RM+) (Tammelin, 2014).

**Proposition 2.2** (Tammelin 2014). *The average external regret of RM+ satisfies  $\text{REG}_{\text{RM+}}(T) \lesssim \sqrt{n/T}$ .*

As alluded to above, RM+ is that (unlike MWU) it is *hyperparameter-free*: there are no step sizes or other hyperparameters to tune. Similarly, RM+ is also *scale-invariant*: if given utility sequence  $\mathbf{u}^1, \dots, \mathbf{u}^T$ , it would produce the same iterates as if it had been given  $C\mathbf{u}^1, \dots, C\mathbf{u}^T$  for any constant  $C > 0$ . These properties make RM+ extremely powerful in practice. In particular, despite a worse theoretical dependence on  $n$ , RM+ is almost always practically superior to MWU. Therefore, we will use it in almost all our experiments.

**Predictive (Optimistic) Algorithms.** *Predictions* can be used to speed up regret minimization algorithms even further. In essence, predictive algorithms take an additional input on every timestep  $t$ , which is a *prediction*  $\tilde{\mathbf{u}}^t$  of the utility vector that it will observe. The algorithm then uses the predicted utility vector to perform a temporary update before returning its strategy. The predictive variants of MWU and RM+ are known respectively as *optimistic multiplicative weights* (OMWU, Chiang et al. 2012; Rakhlin and Sridharan 2013a,b; Syrgkanis et al. 2015) and *predictive regret matching plus* (PRM+, Farina et al. 2021c).<sup>10</sup> Note that by setting  $\tilde{\mathbf{u}}^t = \mathbf{0}$ , the predictive variants collapse to the non-predictive variants. Conventionally (*i.e.*, unless otherwise stated), the prediction is set to the previous observed utility, that is,  $\tilde{\mathbf{u}}^t = \mathbf{u}^{t-1}$ .

Predictive regret matching has the same worst-case guarantee as non-predictive regret matching, but can be significantly faster, both in theory and in practice, if the predictions are accurate.

<sup>10</sup>We use different wording (optimistic vs predictive) to be consistent with usage of past authors.

---

**Algorithm CFR:** Counterfactual regret minimization on tree-form decision problems  $\mathcal{T}$ . For each decision point  $j$ ,  $\mathcal{R}_j$  is a regret minimizer on  $\Delta(\mathcal{A}(j))$ .

---

```

1: initialize  $t \leftarrow 0$ 
2: procedure NEXTSTRATEGY()
3:    $t \leftarrow t + 1$ 
4:    $\mathbf{x}^t[\emptyset] \leftarrow 1$ 
5:   for each decision point  $j$ , in top-down order do
6:      $\mathbf{r}_j^t \leftarrow \mathcal{R}_j.\text{NEXTSTRATEGY}()$ 
7:      $\mathbf{x}^t[j^*] \leftarrow \mathbf{x}^t[p_j]\mathbf{r}_j^t$ 
8:   return  $\mathbf{x}^t$ 
9: procedure OBSERVEUTILITY( $\mathbf{u}^t$ )
10:   $\mathbf{v}^t \leftarrow \mathbf{u}^t$ 
11:  for each decision point  $j$ , in bottom-up order do
12:     $\mathcal{R}_j.\text{OBSERVEUTILITY}(\mathbf{v}^t[j^*])$ 
13:     $\mathbf{v}^t[p_j] \leftarrow \mathbf{v}^t[p_j] + \langle \mathbf{r}_j^t, \mathbf{v}^t[j^*] \rangle$ 

```

---

### 2.3.2 Counterfactual Regret Minimization (CFR)

In this subsection, we will introduce *counterfactual regret minimization* (Zinkevich et al., 2007), following the more recent exposition of Farina et al. (2019a). Intuitively, CFR allows one to *build* a regret minimizer on a tree-form strategy set  $\mathcal{X}$  by running *local* regret minimizers at each decision point, and combining them in a clever way. The guarantee given by CFR can be expressed as follows. Call a subset  $S \subseteq \mathcal{J}$  *playable* if there is a pure strategy that reaches every decision point in  $S$ , that is, there is a pure strategy  $\mathbf{x} \in \mathcal{X}$  such that  $\mathbf{x}[p_j] = 1$  for every  $j \in S$ . Then:

**Proposition 2.3** (Zinkevich et al. 2007; Farina et al. 2019a). *The average external regret of CFR satisfies*

$$\text{REG}_{\text{CFR}}(T) \leq \max_P \sum_{j \in P} \text{REG}_j(T) \leq \sum_{j \in \mathcal{J}} \text{REG}_j(T)$$

where the max is taken over all playable sets  $P$ , and  $\text{REG}_j(T)$  is the regret of the local regret minimizer at decision point  $j$ .

In particular, with (O)MWU and (P)RM+ as the regret minimizers, we get the respective regret bounds

$$\text{REG}_{\text{CFR-(O)MWU}}(T) \lesssim |\mathcal{J}| \sqrt{\frac{\log b}{T}} \leq \frac{|\Sigma|}{\sqrt{T}} \quad \text{and} \quad \text{REG}_{\text{CFR-(P)RM+}}(T) \lesssim |\mathcal{J}| \sqrt{\frac{b}{T}} \leq \frac{|\Sigma|}{\sqrt{T}}$$

where  $b$  is the branching factor.

Several variants of CFR with specific choices of local regret minimizer have special common names. In particular, CFR with RM+ or PRM+ is known as CFR+ or PCFR+ respectively. The latter is currently the fastest regret minimizer in practice in most settings, including game solving (Farina et al., 2021a)<sup>11</sup>.

<sup>11</sup>A notable exception is poker and variants thereof, where *discounted CFR* (Brown and Sandholm, 2019a), which we will not need for this thesis, is sometimes faster.

### 2.3.3 Relation to Equilibrium Finding

There is a well-known, tight connection between no-regret learning and equilibria in games. In particular, we have the following folk result whose proof follows almost directly from the definitions of NFCCE and regret:

**Proposition 2.4.** *In any game, if all players run no-regret learning algorithms over their strategy sets  $\mathcal{X}_i$  with utilities  $u^t(\mathbf{x}_i) := u_i(\mathbf{x}_i, \mathbf{x}_{-i}^t)$ , then after  $T$  rounds, the correlated average strategy profile  $\pi := \text{unif}(\{\mathbf{x}^1, \dots, \mathbf{x}^T\})$  is an  $\epsilon$ -NFCCE, where  $\epsilon = \max_{i \in [n]} \text{REG}_i(T)$  and  $\text{REG}_i(T)$  is the external regret of player  $i$ .*

In zero-sum games, using the fact that NFCCEs collapse to Nash, we have the following analogous result.

**Proposition 2.5.** *In any zero-sum game, if both players run no-regret learning algorithms, then after  $T$  rounds, the uncorrelated average strategy profile  $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ , where  $\bar{\mathbf{x}} = \frac{1}{T} \sum_{t=1}^T \mathbf{x}^t$  (and analogous for  $\bar{\mathbf{y}}$ ) is an  $\epsilon$ -equilibrium, where  $\epsilon = \text{REG}_\blacktriangle(T) + \text{REG}_\blacktriangledown(T)$ .*

Any no-regret learning algorithm for zero-sum games can be run with either *simultaneous* or *alternating* updates. While the above theoretical results apply only to the simultaneous versions, certain algorithms are also known to converge with alternating updates<sup>12</sup>.

---

<sup>12</sup>For example, this is known to be true for CFR+ (Burch et al., 2019), but is nontrivial to show: the original proof attempt by Tammelin et al. (2015) was flawed.

# Part I

# Equilibrium Computation in Adversarial Team Games

## 3 Efficient Parameterized Representations for Team and Imperfect-Recall Equilibrium Computation

### 3.1 Introduction

In two-player zero-sum games, *Nash equilibria in mixed strategies* are the most natural solution concept for modeling rational value-maximizing players. Mixed strategies specify the behavior of a player as a distribution over *pure (deterministic) strategies*. However, the exponential number of such strategies makes the computation of Nash equilibria potentially inefficient. A key assumption to circumvent this issue is *perfect recall*. In a perfect-recall game, the players never forget previously received information or played actions. When this assumption is satisfied,

1. *Kuhn’s theorem* (Kuhn, 1950a) states that *mixed strategies* are equivalent to *behavioral strategies*, which are the strategies expressible as a product of distributions over actions at each decision point.
2. The *sequence-form representation* (von Stengel, 1996; Romanovskii, 1962) of the strategy spaces enables efficient computation of Nash equilibria via a wide variety of different methods. In particular, uncoupled learning dynamics such as CFR converge to a Nash equilibrium by employing a regret minimizer at each decision point of the strategy tree.

There have been significant recent speed improvements to CFR-based techniques (Tammelin et al., 2015; Brown and Sandholm, 2019a; Farina et al., 2021c; Zhang et al., 2024e), and other techniques have been built on top of CFR-based techniques, for example, abstraction algorithms (Sandholm, 2015a,b), subgame solving (Gilpin and Sandholm, 2006; Ganzfried and Sandholm, 2015a; Moravcik et al., 2016; Brown and Sandholm, 2017; Moravčík et al., 2017; Brown and Sandholm, 2018, 2019b), further enhancing scalability. Notable results on large-scale games include poker (Bowling et al., 2015; Moravčík et al., 2017; Brown and Sandholm, 2018, 2019b), Stratego (Perolat et al., 2022), and Diplomacy (, FAIR).

This work seeks to extend these techniques beyond the perfect-recall two-player zero-sum setting. In particular, we focus on computing mixed Nash equilibria in the two equivalent settings of *imperfect-recall games* and *adversarial team games*<sup>13</sup>, for which it is known that computing a Nash equilibrium is NP-hard (Koller and Megiddo, 1992).

Two-player zero-sum imperfect-recall games are characterized by players who may forget information at some point in the game. In this case, a mixed strategy corresponds to a distribution over pure strategies, while a behavioral strategy corresponds to a distribution that performs an independent sampling procedure at each decision point. Unlike for perfect-recall games, Kuhn’s theorem does not apply in imperfect-recall games: mixed strategies can in general be more expressive than behavioral strategies. Imperfect-recall games have been employed in the literature to compress a game representation through forgetfulness (this is the case of some abstraction techniques (Vaugh, 2009; Lanctot et al., 2012; Kroer and Sandholm, 2016)), or by considering human-like agents with imperfect memories (Camerer, 2003).

---

<sup>13</sup>This equivalence is formalized in Section 3.2.2.

Adversarial team games portray two teams of agents facing adversarially. Each team member has utilities identical to her teammates and opposite to members of the opposing team. Effective team coordination is a non-trivial challenge in this setting because team members may have different imperfect information about the current node and no communication channels are available during the game. Intuitively, the player cannot distinguish nodes that are different due to private information revealed to a teammate (such as private cards revealed to them solely). In this case, mixed strategies correspond to strategies coordinated *before the start of the game* through *ex-ante coordination*, while behavioral strategies represent strategies that are not coordinated, in the sense that each agent samples their actions independently from other teammates. Recreational and non-recreational examples of team games include Bridge, security games with multiple defenders and attackers (Jiang et al., 2013), and poker with colluding agents.

Overall, team games are a more common application setting than imperfect-recall games, have many competing works in the equilibrium computation literature, and allow a more intuitive game description. On the other hand, imperfect-recall games yield a cleaner formalism. As these two perspectives are equivalent for our purposes, we choose to adopt an imperfect-recall perspective throughout the rest of the paper to simplify the notation, while using team games to make more intuitive examples for some of the notions introduced.

The main objective of this paper is to propose a novel representation for team and imperfect-recall games by constructing an equivalent two-player zero-sum perfect-recall game. This enables the use of all the solving techniques previously developed for perfect-recall two-player zero-sum games.

We now summarize the contributions of the paper. In Sections 3.3 and 3.3.1, we present an algorithm that converts any two-player zero-sum imperfect-recall game into a strategically-equivalent *perfect-recall game* which we call the *belief game*. We formally prove the equivalence between the two games, and in Section 3.3.2 we show worst-case bounds on the size of the belief game in terms of the number of histories of the original game. In particular, we show that the worst case the number of histories of the belief game is  $O(b^{dk})$ , where  $b$  is the maximum branching factor of the original game,  $d$  is its depth, and  $k$  is a parameter we introduce called the *information complexity*, which intuitively measures the amount of information that can be forgotten by the player—or, in the case of team games, the amount of *information asymmetry* between players on the team.

In Section 3.4, we introduce a notion of DAG-form decision-making that we use to generalize counterfactual regret minimization (CFR) beyond tree-form games. While we introduce it for the purpose of applying it to imperfect-recall games, we believe it to be of independent interest as well.

In Section 3.5, we use DAG-form decision problems to efficiently represent each player’s strategy space in the belief game through a construction we call the *team-belief DAG* (TB-DAG). We show that the TB-DAG representation of a game with imperfect recall can be exponentially smaller than the size of the belief game and that it can be constructed directly from the original game without first constructing the belief game, thus leading to exponentially faster algorithms in the worst case. This construction improves the worst-case efficiency<sup>14</sup> of our technique to  $O(|\mathcal{H}|(b+1)^{k+1})$ , where  $|\mathcal{H}|$  is the number of nodes in the original game. We also show that this bound is essentially optimal: under reasonable computational assumptions (namely, the exponential time hypothesis), we show that there cannot exist an algorithm for solving even single-player games of imperfect recall whose runtime is of the form  $f(k)\text{poly}(|\mathcal{H}|)$ , for *any* function  $f$ .

In Section 3.6 we investigate the computational complexity of computing mixed Nash equilibria with imperfect recall. We prove that computing a max-min strategy in mixed or behavioral strategies in games where both players have imperfect recall is  $\Delta_2^P$ -complete and  $\Sigma_2^P$ -complete respectively.

Section 3.7 presents further discussions comparing different notions presented in the paper, providing further insights on the technical decisions made.

In Section 3.8, we evaluate our methods empirically by benchmarking our construction on a standard testbed of imperfect-information games, compared to state-of-the-art baselines. We find that our technique allows much faster equilibrium computation when the information complexity  $k$  of the game is low.

<sup>14</sup>By *efficiency* here we mean the size of the representation of the strategy spaces of the players. Algorithms such as CFR have per-iteration complexity that scales linearly in this size.

We have defined equilibrium concepts for team games by using an “equivalent” coordinator game that is two-player zero-sum imperfect recall. It turns out that, in fact, every two-player zero-sum imperfect-recall game  $\Gamma'$  has an ATG whose coordinator game is  $\Gamma'$ : indeed, given such a  $\Gamma'$ , consider the ATG  $\Gamma$  in which every information set is assigned to a different player. Therefore, team games and imperfect-recall games are in a very strong sense equivalent. All of the results of this section, unless otherwise stated, therefore apply equally to team games and to two-player zero-sum imperfect-recall games.

In this section, we opt to consider the point of view of two-player zero sum games with imperfect recall. A summary of the different equivalent terms that are used in the two settings can be found in Table 2.

We introduce the fundamental contribution of the paper: a novel technique to compute a mixed Nash equilibrium in two-player zero-sum imperfect-recall games (or equivalently to compute a TMECoR in adversarial team games) based on the construction of an equivalent two-player zero-sum game with perfect recall.

Our technique attains the perfect-recall condition by suitably changing the information available to the players, as well as their action sets. The main intuition behind the belief game is to consider the point of view of a perfect recall player in place of the imperfect-recall one. Differently from the imperfect-recall player, this player reasons only using information the player would never forget due to imperfect recall and chooses an action for every possible information set the imperfect-recall player may be in. The game then transitions by applying the action corresponding to the information set of the current node. Crucially, the perfect-recall player can strategically refine the set of reached nodes over time by carefully considering reachable nodes given the played strategy and the perfect-recall results of her actions.

After introducing the main concepts and the construction algorithm, we prove that the original and the belief games are strategically equivalent. This means that the perfect-recall player we introduce is an equivalent representation of both the imperfect-recall player and the corresponding preplay coordinated team (thanks to the considerations from Section 3.2.2).

## 3.2 Preliminaries

Since this part deals with equilibrium computation in *team* games and *games with imperfect recall*, we first introduce some notation and definitions that pertain to these. For this part, unless otherwise stated, all games are assumed to be timeable.

### 3.2.1 Behavioral and Mixed Max-Min Strategies

Recall first the definition of a mixed-strategy Nash equilibrium for a game:

**Definition 3.1** (Mixed-strategy Nash equilibrium). In a two-player zero-sum game, a (realization-form) Nash equilibrium is a saddle-point solution to the optimization problem

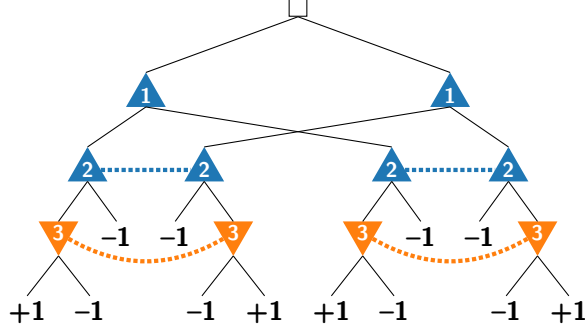
$$\max_{\mathbf{x} \in \text{co } \mathcal{X}} \min_{\mathbf{y} \in \text{co } \mathcal{Y}} u(\mathbf{x}, \mathbf{y}).$$

Since this problem is a bilinear saddle-point problem and  $\text{co } \mathcal{X}$  and  $\text{co } \mathcal{Y}$  are convex, the minimax theorem applies, and the maximization and minimization can be freely swapped without changing the value of the game. The optimal value of the above program is the Nash equilibrium value of the game.

For games with imperfect recall, restricting to *behavioral strategies* is a nontrivial restriction. Recall that a behavioral strategy is a mixed strategy that mixes independently at each information set. Thus, the realization form of a behavioral strategy is obtained by multiplying the probability of picking each action of the player on the  $\emptyset \rightarrow z$  path. We will use  $\hat{\mathcal{X}}_i$  to denote the set of realization-form behavioral strategies of a player  $i$ . Recall that Kuhn’s theorem states that, in games with perfect recall, behavioral and mixed strategies are realization-equivalent. That is,  $\hat{\mathcal{X}}_i = \text{co } \mathcal{X}_i$ .

**Definition 3.2** (Behavioral max-min strategy). In a two-player zero-sum game, a *behavioral max-min strategy*  $\mathbf{x} \in \hat{\mathcal{X}}$  is a solution to the optimization problem

$$\max_{\mathbf{x} \in \hat{\mathcal{X}}} \min_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{x}, \mathbf{y})$$



**Figure 1:** An example of an adversarial team game. There are three players:  $P1$  and  $P2$  are on team  $\Delta$ , and  $P3$  is on team  $\nabla$ . Dotted lines connect nodes in the same information set. The (total) utility of  $\Delta$  is listed on each terminal node. The root node is a nature node, at which nature selects uniformly at random.

The *behavioral max-min value* is the optimal value of the above problem. Since  $\hat{\mathcal{X}}$  and  $\hat{\mathcal{Y}}$  are not necessarily convex sets, the minimax theorem does not apply, so the maximization and minimization can not necessarily be swapped. Therefore—unlike the mixed-strategy Nash—the behavioral max-min strategy is not an *equilibrium*. Further, in games with imperfect recall, the tree-form decision problem is not a valid representation of the set of realization-form strategies. Therefore, we will need different techniques to tackle such games.

### 3.2.2 Adversarial Team Games

The general framework of adversarial team games has first been studied by von Stengel and Koller (1997) in the context of normal form games, while Celli and Gatti (2018) first addressed them in an extensive-form setting. Adversarial team games describe situations where multiple agents are organized in two-teams receiving zero-sum payoffs. The paper focuses on the setting in which no extra communication channel is available to the players during the game, but they are allowed to communicate freely before the start of the game. This means that the only form of coordination across players’ strategies available is *preplay coordination*, *i.e.* any coordination has to be prepared before the start of the game.

Adversarial team games can be modeled as extensive-form games as follows:

**Definition 3.3** (Adversarial team game). An extensive-form, perfect-recall game is said to be an *adversarial team game* (ATG), or *two-team zero-sum game* iff:

- the player set is partitioned in two sets called *teams*, symbolized by  $\Delta$  and  $\nabla$ . Formally,  $[n] = \Delta \cup \nabla$ ;
- the utilities of the players belonging to the same team are identical, and the total utilities of the two team are opposites. Formally:

$$\begin{aligned}
 u_i &= u_j \quad \text{for all } i, j \in \Delta \\
 u_i &= u_j \quad \text{for all } i, j \in \nabla \\
 \sum_{i \in \Delta} u_i &= - \sum_{j \in \nabla} u_j
 \end{aligned}$$

In adversarial team games, the Nash equilibrium fails to take into account the fact that teams can coordinate among themselves. Indeed, it is possible for there to be a Nash equilibrium in which two teammates could profit by *jointly* switching strategies, but no *individual* player can profit from a unilateral deviation. To take into account these joint deviations, it is most natural to reformulate an adversarial team game as a two-player zero-sum game of imperfect recall, in which a *team coordinator* plays on behalf of all members of that team. In this manner, deviations of the team coordinator correspond to *simultaneous, joint* deviations of all team members. We now formalize this conversion.

**Definition 3.4** (Coordinator game). Let  $\Gamma$  be an adversarial team game. The *coordinator game*  $\Gamma'$  corresponding to  $\Gamma$  is the two-player zero-sum imperfect-recall game  $\Gamma'$ , where

$$\mathcal{I}'_{\blacktriangle} = \bigcup_{i \in \Delta} \mathcal{I}_i, \quad \mathcal{I}'_{\blacktriangledown} = \bigcup_{i \in \nabla} \mathcal{I}_i, \quad u'_{\blacktriangle} = \sum_{i \in \Delta} u_i, \quad \text{and} \quad u'_{\blacktriangledown} = \sum_{i \in \nabla} u_i.$$

The coordinator game merges all members of a team ( $\Delta$  or  $\nabla$ ) into a coordinator ( $\blacktriangle$  or  $\blacktriangledown$ ). Therefore:

- Pure strategies of a coordinator correspond to *pure profiles* of the team.
- Behavioral strategies of a coordinator correspond to *behavioral profiles* of the members of the team. Since behavioral strategies enforce actions at different infosets to be independently sampled, this means that team members can *privately* sample randomness for their own personal use but cannot share that randomness with teammates.
- Mixed strategies of a coordinator correspond to *correlated* strategy profiles of the members of the team. In a correlated profile, team members may *jointly* sample randomness that they use to correlate their actions.

We remark on the role that preplay coordination has in allowing the coordination capabilities modeled by the coordinator game. In fact, before starting the game, players are allowed to jointly sample a pure plan from their coordinator’s mixed strategy and then individually play the specified actions at the infoset in which they play. This allows the team to play any randomized strategy of the coordinator effectively.

The coordinator game allows us to define notions of equilibrium specialized for team games:

**Definition 3.5.** A *team max-min equilibrium with correlation* (TMECor) of an ATG  $\Gamma$  is a mixed-strategy Nash equilibrium of  $\Gamma'$ .

**Definition 3.6.** A *team max-min equilibrium* (TME) of an ATG  $\Gamma$  is a behavioral max-min strategy of  $\Gamma'$ .

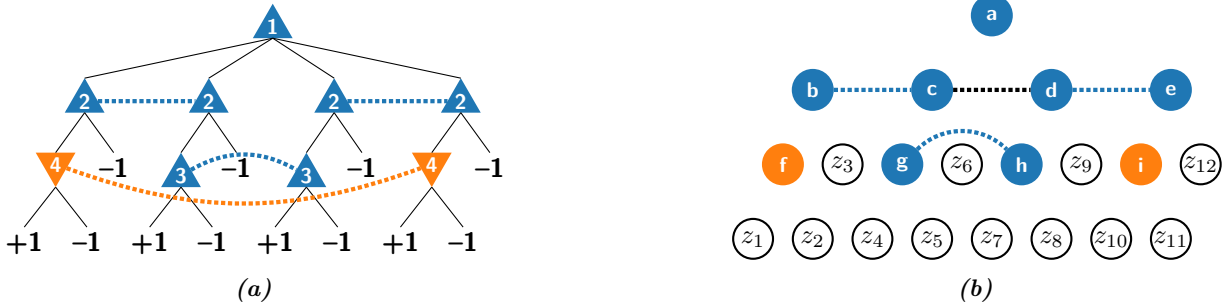
The *TMECor value* and *TME value* are defined analogous to the Nash value and behavioral max-min value. As discussed before, behavioral max-min strategies in  $\Gamma'$  are not equilibria in  $\Gamma'$ , so one may wonder about the name “team max-min equilibrium”. However, there *is* a sense in which TMEs are equilibria: [von Stengel and Koller \(1997\)](#) showed that, at least in the case where  $|\nabla| = 1$ , the TMEs are precisely the Nash equilibria of the team in  $\Gamma$  that maximize the utility of team  $\Delta$ .

An example adversarial team game in which the difference between TME and TMECor is relevant can be found in [Figure 1](#). The coordinator game is constructed simply by erasing the player labels, creating a two-player zero-sum game. This game is a simple signaling game: nature selects a bit, which is privately revealed to P1. P1 then communicates a single bit, which is publicly revealed. Then P2 and P3 both attempt to guess nature’s selected bit, and  $\Delta$  wins if and only if P2’s guess is correct. Therefore, the goal of P1 and P2 is for P1 to “securely” communicate the bit to P2 without also revealing it to P3. With a behavioral profile, this is impossible, since P1 and P2 cannot correlate their strategies; therefore, the TME value is  $-1/2$ . However, if P1 and P2 are allowed to correlate their strategies, they can do the following: jointly flip a coin. If that coin landed heads, P1 communicates the true bit, and P2 plays what P1 communicates. If that coin landed tails, P1 communicates the opposite of the true bit, and P2 plays the opposite of what P1 communicates. In this way, P2 will always play the true bit, but P3 (who does not know the outcome of the correlating coinflip) does not learn any information. Therefore, the value of this strategy for  $\Delta$  is 0 (since P2 wins half the time by randomly guessing the bit).



Adversarial Team Games	Imperfect-Recall Games
Team $\blacktriangle\blacktriangledown$	Player $\blacktriangle\blacktriangledown$
Correlated team strategy	Mixed strategy
Uncorrelated team strategy	Behavioral strategy
TMECor	Mixed-strategy Nash equilibrium
TME	Behavioral max-min strategy

**Table 2:** Translation table between terms commonly employed in the adversarial team games and two-player imperfect recall games. The translation happens through the introduction of coordinator games (Definition 3.4).



**Figure 3:** An example of team game (a) (using the same notation as Figure 1) and its corresponding connectivity graph for player  $\blacktriangle$  (b). Nodes in the two figures correspond in position.

### 3.3 Beliefs and Observations

The main purpose of this section is to formally define *beliefs*, which are sets of nodes  $B \subseteq \mathcal{H}$  derived from information sets  $\mathcal{I}_i$  of player  $i \in \{\blacktriangle, \blacktriangledown\}$ . Informally, beliefs are the “information sets” that player  $i$  would have if she could not distinguish nodes that cannot be distinguished using information from a later stage. This notion is formalized by putting in the same belief any two nodes that have descendent nodes in the same information set (even if they belong to different information sets). Similarly to information sets:

1. nodes in beliefs would be indistinguishable to  $i$ ,
2. one action is chosen at each belief, and then this action is followed in all nodes in the belief, and
3. if the player knows that the current node of the game  $h$  lies in a set  $H$  of candidates, and the player observes that her current belief is  $B$ , then the set of candidates can be refined to  $B \cap H$  (i.e. similarly to information sets, beliefs correspond to observations over the state of the game).

Crucially, beliefs can be organized in the tree-like structure needed by algorithms finding Nash equilibria in two-player zero-sum games, as we will see in Sections 3.3.1 and 3.3.3. This is thanks to the guarantee that once a group of nodes is split among two different distinguishable beliefs, then any group of descendent nodes from one belief will be distinguishable from any group of descendants from the other.

In the following, we formalize the notion of beliefs and observations. We consider a two-player zero-sum game with imperfect recall  $\Gamma$ .

**Connectivity graph.** We say that two nodes  $h$  and  $h'$  are *unforgettably distinguished* by  $i$  if they do not belong to the same infoset and no pair of children of those two nodes belong to the same infoset, *i.e.*  $i$  will never be in an information set where these two ancestors are both possible. This condition guarantees that if the set of candidates is  $H = \{h, h'\}$ , then the player is able to discern  $h$  from  $h'$  and will never forget which of the two nodes has been reached in the next steps of the game.<sup>15</sup>

For the purpose of our definitions, we are concerned with pair of nodes that are *not* distinguishable. This can be represented through a *connectivity graph* over  $\mathcal{H}$  as follows.

**Definition 3.7** (Connectivity graph). The *connectivity graph*  $\mathcal{G}_i = (\mathcal{H}, \mathcal{E}_i)$  for player  $i \in \{\blacktriangle, \blacktriangledown\}$  is the graph with nodes  $\mathcal{H}$  and edges  $\mathcal{E}_i$ , where  $(h, h') \in \mathcal{E}_i$  if  $h$  and  $h'$  are at the same depth in  $\Gamma$  and there exists  $I \in \mathcal{I}_i$  such that  $h \preceq I$  and  $h' \preceq I$ .

Consider Figure 3b as an example of connectivity graph for a game. Note the blue edges, which correspond to connections due to infosets, and the black edge  $c - d$  due to  $g, h$  belonging to the same infoset.

**Beliefs.** Consider now a set  $H$  of nodes such that the induced subgraph  $\mathcal{G}_i[H]$  is connected. Player  $i$  has no way of distinguishing any subset of  $H$  from the others, because any node cannot be distinguished from its neighbors. *Beliefs* are defined as these sets of indistinguishable nodes.<sup>16</sup>

**Definition 3.8** (Belief). A set of nodes  $B \subseteq \mathcal{H}$  is a *belief* for player  $i$  if the induced subgraph  $\mathcal{G}_i[B]$  is connected.

We remark that the timeability property assumed on  $\Gamma$  implies that any node belonging to the same belief has the same depth. Notice that a direct consequence of the definition of beliefs is that  $\{\emptyset\}$  and  $\{z\}$  for  $z \in \mathcal{Z}$  are singleton beliefs for both teams.

**Observations.** Consider instead a set  $H$  of nodes such that the induced subgraph  $\mathcal{G}_i[B]$  has different connected components. In this case, player  $i$  can distinguish those components one from the other, thus partitioning  $H$  into multiple beliefs. Intuitively, the unforgettable information is enough to distinguish every node in a component from any node in other components. The player can, therefore, exclude nodes from components that are distinguishable from the current reached node. We say that upon reaching a node  $h$  among possible candidates  $H$ , player  $i$  *observes belief*  $B \subseteq H$ , meaning that player  $i$  uses the newly acquired unforgettable information acquired in  $h$  to refine its imperfect information from  $H$  to  $B$ . We formalize this notion of observation through the function  $\text{SPLITBELIEF}_i$ .<sup>17</sup>

**Definition 3.9** (Observation). The *observation* for player  $i \in \{\blacktriangle, \blacktriangledown\}$  when reaching node  $h$  among a set  $H$  of candidate nodes is:

$$\text{SPLITBELIEF}_i(H, h) := \text{the connected component of } \mathcal{G}_i[H] \text{ containing } h.$$

The set of all possible observations given a set of candidates is denoted<sup>18</sup> by

$$\mathcal{B}_H := \{\text{SPLITBELIEF}_i(H, h) : h \in H\}.$$

An example of observation can be given by considering the team game depicted in Figure 3 and a candidate set  $H = \{b, c, e\}$ . This candidate set is possible when player  $\blacktriangle$  plays a strategy where player 1 plays a mixed

<sup>15</sup> $h, h'$  being distinguishable implies that in the corresponding team game any team member can recall whether  $h$  or  $h'$  was reached upon reaching  $h$  or  $h'$ .

<sup>16</sup>From a team game perspective, beliefs are sets of nodes with the guarantee that once reached all team members know that any node  $\mathcal{H} \setminus B$  is not reached, *i.e.* it is team-common knowledge that the game reached a node in  $B$ .

<sup>17</sup>In team games, the belief returned by  $\text{SPLITBELIEF}_i$  is the team-common knowledge update happening when reaching  $h$  among a set of candidates  $H$ .

<sup>18</sup>We remark that the belief-based constructions employed by the paper would also work when allowing  $\text{SPLITBELIEF}_i$  to return any superset of connected components. For example, in the framework of *factored-observation games* (Kovařík et al., 2022), it is valid to define  $\text{SPLITBELIEF}_i$  using the explicitly-given public observations. However, since the efficiency of the proposed algorithms depends on the size of the beliefs employed, we opt not to allow, by definition, the use of beliefs larger than needed. As we show in Section 3.3.2, any reduction in the size of the beliefs in a game brings exponential benefits in the size of the belief game obtained.

strategy excluding  $d$  from its support. This, in turn, implies that player 2, at the next step, knows that the reached node  $h$  in the game is in  $H$ . Moreover, player 2 observes her current information set  $I = b, c$  if  $h \in \{b, c\}$  of  $I = \{d, e\}$  if  $h \in \{d, e\}$ .  $I$  can be used to further refine  $H$  as long as the information used will be known at player 1 next. This is formalized in  $\text{SPLITBELIEF}_i(H, b) = \text{SPLITBELIEF}_i(H, c) = \{b, c\}$  and  $\text{SPLITBELIEF}_i(H, e) = \{e\}$ , which intuitively correspond to the fact that given those candidates,  $\blacktriangle$  unforgettably distinguishes  $b$  and  $c$  from  $e$ . From the equivalent team game perspective: player 2 is active and can check her current infoset to distinguish the two beliefs; player 1 has stopped playing and therefore it is not relevant in terms of team knowledge; player 3 either will not play or will know that the current node was  $c$  once the game reached  $g$ , so she can safely assume that the game is in  $c$ . This means that every player distinguishes  $e$  from  $b, c$ .

**Team public states.** We compare our notion of beliefs with *public states*, an alternative customarily used in the related literature. A public state  $P$  for player  $i$  is a connected component of the connectivity graph  $\mathcal{G}_i$ . The set of all public states of  $i$  is denoted as  $\mathcal{P}_i$ .

Public states identify sets of nodes that are distinguishable to a player without considering a possibly pruned subgraph of  $\mathcal{G}_i$  as instead done for team observations. Therefore, every belief is contained in a public state. In Figure 3 we have that  $\mathcal{P}_{\blacktriangle} = \{\{a\}, \{b, c, d, e\}, \{g, h\}, \{f\}, \{i\}\} \cup \{\{z\} : z \in \mathcal{Z}\}$ .

Public states are the customarily adopted alternative to observations when partitioning a set  $H$  of candidates in beliefs by splitting  $H$  in  $\{H \cap P : P \in \mathcal{P}_i\}$ . However, public states may return a coarser partition than the one returned by observations, as the absence of specific nodes from  $H$  may disconnect components in  $\mathcal{G}$ . We will, therefore, use observations in place of public states whenever possible. An example illustrating the difference between the two definitions is available in Section 3.7.1.

**Prescriptions.** Restricting the information available to player  $i$  to her beliefs also affects the set of actions available. In fact, multiple infosets may intersect a given belief, and the player does not know in which infoset she finds herself. Therefore, she does not know what actions are available to her.

We overcome this issue by associating to each belief  $B$  a set of meta-actions  $\mathcal{A}_i(B)$  such that an action is specified for each possible infoset that intersects the belief. We call such structured meta-actions *prescriptions* and use a symbol  $\mathbf{a}$  to indicate them. The concept is formally defined as follows.

**Definition 3.10** (Prescription). Consider a belief  $B$  of a player  $i \in \{\blacktriangle, \blacktriangledown\}$ . A *prescription*  $\mathbf{a}$  is a selection of one action at each infoset having a nonempty intersection with  $B$ :

$$\mathbf{a} \in \prod_{I \in \mathcal{I}_i[B]} \mathcal{A}(I) \quad \text{where} \quad \mathcal{I}_i[B] = \{I \in \mathcal{I}_i : I \cap B \neq \emptyset\}.$$

Given a prescription  $\mathbf{a}$  for a belief  $B$  and an infoset  $I$  such that  $I \cap B \neq \emptyset$ , we denote as  $\mathbf{a}[I]$  the action relative to infoset  $I$  which is specified by prescription  $\mathbf{a}$ . Note that we have *empty prescriptions* at beliefs containing no active nodes for a player.

As we will see in the next section, our equivalent belief game introduces one perfect-recall player per team, with information sets associated with beliefs corresponding to the perfect-recall part of the information available to this unique player. Prescriptions will allow this player to have an identical expressive power in terms of actions without accessing the exact information set of the player, which is her imperfect-recall information. Moreover, specifying a prescription at each reached belief for  $i$  incrementally defines a pure strategy of player  $i$ . This allows us to consider a reduced set of candidate nodes  $H$  for the reached node  $h$  from which the belief is observed, as a non-played action implies that all the descendant nodes are not reached and, therefore, excluded from the candidates.

For example, consider a 3-player poker instance where two players collude to form a team. At any time of the game, we can consider the point of view of a team coordinator, who acts as the single imperfect recall player. We can imagine this coordinator as sitting at the same table as the players, and therefore, she cannot access the private cards given to the players but can access the same public information as the players, that is, the bet, fold, and check actions of the players. Her belief at any point regards the private cards that each

team member has. At the start of the game, this belief is uniform over all pairs of cards, as no information regarding these cards is available from an external point of view. The coordinator emits prescriptions for the players to follow as the game progresses. Since the coordinator does not know the card held by a player, she has to prescribe an action for each possible card the current player may hold. The player receives this prescription and follows the part of it that matches the private card. By observing the action played by the player, the coordinator can exclude from her belief the cards for which she prescribed different actions. While there are no means of communicating prescriptions during play at the poker table, this mechanism can be implemented *ex ante*; that is, each team of players jointly samples a pure strategy of this coordinator before the start of the game, and each member simulates the coordinator locally.

**Information complexity.** We quantify the number of information sets reaching a belief through the notion of *information complexity*  $k$ . This quantity will allow us to bound the size of the belief games in Sections 3.3.2 and 3.5.1.

We first characterize the notion of *remembered* information sets and the set of *last-infosets* at a node. Intuitively, an infoset  $J$  remembers another infoset  $I$  if reaching a node in  $J$  implies traversing a node in  $I$  and picking a specific action. Therefore, knowing to be at a node in  $J$  allows the player to recall having traversed  $I$  and have played action  $a$  there. The last-infosets of player  $i$  at  $h$  are the information sets traversed by  $h$  and not remembered by any following information set of the player up to  $h$ .<sup>19</sup> This set quantifies the knowledge lost by the player at a node due to imperfect recall.

**Definition 3.11.** An infoset  $J$  *remembers* another infoset  $I$  if there exists an action  $a \in \mathcal{A}(I)$  such that, for every  $h \in J$ , we have  $h'a \preceq h$  for some  $h' \in I$ .

**Definition 3.12.** The set of *last-infosets* at node  $h$  for player  $i$  is the set of infosets  $I \in \mathcal{I}_i$  such that  $I \preceq h$  and there is no other infoset  $J \in \mathcal{I}_i$  such that  $J \preceq h$  and  $J$  remembers  $I$ .

We will use  $\text{LI}_i(h)$  to denote the set of last-infosets at  $h$  for player  $i$ . Note that if  $h \in \mathcal{H}_i$  then  $I_h \in \text{LI}_i(h)$ .

Now define the *information complexity*  $k$  of a two-player game  $\Gamma$  as follows.

$$k = \max_{\substack{i \in \{\blacktriangle, \blacktriangledown\}, \\ P \in \mathcal{P}_i}} \left| \bigcup_{h \in P} \text{LI}_i(h) \right|$$

Intuitively,  $k$  is a representation of *how much information can be worst-case forgotten* by player  $i$ . In the team game interpretation,  $k$  is a representation of how *asymmetric* the information is among team members. Note that  $k = 1$  if and only if both players have perfect recall.

The information complexity characterizes both the number of beliefs in a public state  $P$ , and the number of prescriptions that are available at such beliefs. In fact, the actions played at information sets in  $\bigcup_{h \in P} \text{LI}_i(h)$  determine which nodes in  $P$  are reached (that is, a belief  $B \subseteq P$ ).

As an example, consider the game from Figure 3 and the public state  $P = \{g, h\}$ . We have that the strategy played at

$$\bigcup_{h \in \{g, h\}} \text{LI}_i(h) = \{I_a, I_c, I_g\} \cup \{I_a, I_d, I_h\} = \{I_a, I_c, I_d, I_g\}$$

is enough to characterize a belief  $B \in P$  and a prescription at that belief. In fact, the action at  $I_a$  decides whether  $c$  and  $d$  are reached, the actions at  $I_c$  (respectively  $I_d$ ) decide whether  $g$  (respectively  $h$ ) is reached, and the action at  $I_g = I_h$  is the prescription.

It is instructive to understand how  $k$  behaves in a simple game. Suppose that  $\Gamma$  is a team game such that there are  $n$  players on each team, each player is assigned one of  $t$  “private types” (in poker, these are the private hands) and all other information in the game is common knowledge. Then at each public state  $P \in \mathcal{P}_i$ , there are at most  $t$  last-infosets per player, so  $k = nt$ .

<sup>19</sup>From the perspective of an adversarial team game, the last-infosets at a node for team  $t \in \{\blacktriangle, \blacktriangledown\}$  are the most recent infosets of each player in  $t$ , minus the infosets of players that are implied by other players’ infosets.

---

**Algorithm MakeBeliefGame:** Belief game construction

---

```
1: procedure MAKENODE $\blacktriangle$ ( $h, B_{\blacktriangle}, B_{\blacktriangledown}, \tilde{\sigma}_{\blacktriangle}, \tilde{\sigma}_{\blacktriangledown}$ )
2:   create node  $\tilde{h} \in \tilde{\mathcal{H}}_{\blacktriangle}$ 
3:   add  $\tilde{h}$  to infoset labeled  $(\tilde{\sigma}_{\blacktriangle}, B_{\blacktriangle})$ 
4:   for each prescription  $\mathbf{a}_{\blacktriangle} \in \mathcal{A}_{\blacktriangle}(B_{\blacktriangle})$  do
5:      $\tilde{h}\mathbf{a}_{\blacktriangle} \leftarrow \text{MAKENODE}_{\blacktriangledown}(h, B_{\blacktriangle}, B_{\blacktriangledown}, \tilde{\sigma}_{\blacktriangle}, \tilde{\sigma}_{\blacktriangledown}, \mathbf{a}_{\blacktriangle})$ 
6:   return  $\tilde{h}$ 
7: procedure MAKENODE $\blacktriangledown$ ( $h, B_{\blacktriangle}, B_{\blacktriangledown}, \tilde{\sigma}_{\blacktriangle}, \tilde{\sigma}_{\blacktriangledown}, \mathbf{a}_{\blacktriangle}$ )
8:   create node  $\tilde{h}\mathbf{a}_{\blacktriangle} \in \tilde{\mathcal{H}}_{\blacktriangledown}$ 
9:   add  $\tilde{h}\mathbf{a}_{\blacktriangle}$  to infoset labeled  $(\tilde{\sigma}_{\blacktriangledown}, B_{\blacktriangledown})$ 
10:  for each prescription  $\mathbf{a}_{\blacktriangledown} \in \mathcal{A}_{\blacktriangledown}(B_{\blacktriangledown})$  do
11:     $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown} \leftarrow \text{MAKENODE}_{\mathcal{C}}(h, B_{\blacktriangle}, B_{\blacktriangledown}, \tilde{\sigma}_{\blacktriangle}, \tilde{\sigma}_{\blacktriangledown}, \mathbf{a}_{\blacktriangle}, \mathbf{a}_{\blacktriangledown})$ 
12:  return  $\tilde{h}\mathbf{a}_{\blacktriangle}$ 
13: procedure MAKENODE $\mathcal{C}$ ( $h, B_{\blacktriangle}, B_{\blacktriangledown}, \tilde{\sigma}_{\blacktriangle}, \tilde{\sigma}_{\blacktriangledown}, \mathbf{a}_{\blacktriangle}, \mathbf{a}_{\blacktriangledown}$ )
14:  if  $h$  is terminal node then
15:    create new terminal node  $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown} \in \tilde{\mathcal{Z}}$ 
16:     $u_i(\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown}) \leftarrow u_i(h)$  for each player  $i$ 
17:     $\mathbf{p}[\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown}] \leftarrow \mathbf{p}[h]$ 
18:    return  $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown}$ 
19:  create new chance node  $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown} \in \mathcal{H}_{\mathcal{C}}$ 
20:  if  $h$  is a chance node then  $S \leftarrow \{ha : a \in \mathcal{A}(h)\}$ 
21:  else  $S \leftarrow \{ha_i[I_h]\}$  where  $h \in \mathcal{H}_i$ 
22:  for each node  $ha \in S$  do
23:     $B'_i \leftarrow \text{SPLITBELIEF}_i(B_i\mathbf{a}_i, ha)$  for each player  $i$ 
24:     $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown} \leftarrow \text{MAKENODE}_{\blacktriangle}(ha, B_{\blacktriangle}', B_{\blacktriangledown}', \tilde{\sigma}_{\blacktriangle} + (\tilde{\sigma}_{\blacktriangle}, \mathbf{a}_{\blacktriangle}), \tilde{\sigma}_{\blacktriangledown} + (\tilde{\sigma}_{\blacktriangledown}, \mathbf{a}_{\blacktriangledown}))$ 
25:  return  $\tilde{h}\mathbf{a}_{\blacktriangle}\mathbf{a}_{\blacktriangledown}$ 
```

---

### 3.3.1 Belief Game Construction

We now introduce an algorithm that explicitly constructs a belief game given any two-player game. We will use  $\tilde{\Gamma}$  to denote the belief game and distinguish components of the original game  $\Gamma$  from components of the belief game by writing tildes: for example, a generic history is  $\tilde{h} \in \tilde{\mathcal{H}}$ , a generic information set is  $\tilde{I}_i \in \tilde{\mathcal{I}}_i$ , and so on.

Here, for cleanliness, we will describe the evolution of the belief game as a game of *simultaneous moves*. Algorithm **MakeBeliefGame** describes the procedure that constructs an extensive-form game (without simultaneous moves) that is equivalent to it.<sup>20</sup> In particular,  $\text{MAKENODE}_{\blacktriangle}(\emptyset, \{\emptyset\}, \{\emptyset\}, \emptyset, \emptyset)$  constructs the whole belief game.

A node  $\tilde{h} \in \tilde{\mathcal{H}}$  in the belief game is identified by a tuple  $(h, B_{\blacktriangle}, B_{\blacktriangledown})$  such that  $h \in B_{\blacktriangle} \cap B_{\blacktriangledown}$ , where  $h \in \mathcal{H}$  is the corresponding node in the original game describing the underlying state of the game,  $B_{\blacktriangle}, B_{\blacktriangledown}$  are the current beliefs of  $\blacktriangle$  and  $\blacktriangledown$  respectively. At  $\tilde{h}$ , each player  $i \in \{\blacktriangle, \blacktriangledown\}$  has a (possibly empty) collection of infosets,  $\mathcal{I}[B_i]$ , at which it needs to prescribe an action. The two players simultaneously submit actions  $\mathbf{a}_i \in \mathcal{A}_i(B_i)$ . The next belief game node is  $(ha, B'_{\blacktriangle}, B'_{\blacktriangledown})$ , where:

- (i) The action  $a$  is the one taken by the player at  $h$ : if  $h$  is chance node, then  $a$  is sampled from chance's action distribution at  $h$ ; otherwise,  $a = \mathbf{a}_i[I_h]$ .
- (ii) the beliefs evolve as follows. For each player  $i$ , the set of candidate next histories in the original game

---

<sup>20</sup>We implement simultaneous actions by representing each step in the game as a sequence of one node per player  $\blacktriangle, \blacktriangledown, \mathcal{C}$  where everyone acts; the effects of the actions taken are applied at the end.

compatible with  $i$ 's current belief  $B_i$  and its prescription  $\mathbf{a}_i$  is given by

$$B_i \mathbf{a}_i := \underbrace{\{ha : h \in B_\blacktriangle \cap \mathcal{H}_i, a = \mathbf{a}[I_h]\}}_{\substack{\text{when player } i \text{ acts,} \\ \text{it must be according to the prescription}}} \cup \underbrace{\{ha : h \in B_\blacktriangle \setminus \mathcal{H}_i, a \in \mathcal{A}(h)\}}_{\substack{\text{when player } i \text{ does not act,} \\ i \text{ does not know what action is taken}},$$

Next, player  $i$  observes the information revealed by the next history  $ha$ , thus arriving at belief

$$\tilde{B}'_i := \text{SPLITBELIEF}_i(B_i \mathbf{a}_i, ha).$$

We remark some characteristics of  $\tilde{\Gamma} := \text{MakeBeliefGame}(\Gamma)$ .

- Multiple different tree nodes  $\tilde{h}$  can correspond to the same  $(h, B_\blacktriangle, B_\blacktriangledown)$  tuple. In particular, for each terminal node  $z \in \mathcal{Z}$  there is only one state  $(z, \{z\}, \{z\})$ .
- Information sets in  $\tilde{\Gamma}$  are associated to sequences of beliefs and prescriptions. In particular, such infosets can be described by tuples of the form  $(B_i^1 = \{\emptyset\}, \mathbf{a}_i^1, B_i^2, \mathbf{a}_i^2, \dots, B_i^L)$ , where  $\mathbf{a}_i^\ell \in \mathcal{A}(B_i^\ell)$  and  $B_i^{\ell+1} = \text{SPLITBELIEF}_i(B_i^\ell \mathbf{a}_i^\ell, h)$  for some  $h \in B_i^\ell \mathbf{a}_i^\ell$ .
- By construction of **MakeBeliefGame** we have that  $\tilde{\Gamma}$  is a perfect-recall game. In fact, nodes with different sequences are associated to different information sets thanks to including sequences in each information set's label;
- When  $h$  is terminal, the belief game does not stop until both players have observed the trivial belief  $\{h\}$  at  $h$  and then submitted their empty prescriptions at that belief. This is for notational convenience: it ensures that terminal sequences for a player  $i$  will always end with singleton beliefs, which will make the later analysis cleaner.
- Modulo trivial reformulations (namely, the insertion of nodes with a single child), if  $\Gamma$  is perfect recall then  $\tilde{\Gamma}$  is identical to  $\Gamma$ .

Given a pure strategy  $\tilde{\mathbf{x}}_i \in \tilde{\mathcal{X}}_i$ , we say that  $\tilde{\mathbf{x}}_i$  plays to a belief  $B_i$  of player  $i$  if  $\tilde{\mathbf{x}}_i$  plays to some node corresponding to  $(h, B_i, B_{-i})$ .

**Theorem 3.13.** *Let  $\Gamma$  be any two-player imperfect-recall extensive-form game, and  $\tilde{\Gamma}$  be the belief game constructed by **MakeBeliefGame**.  $\Gamma$  and  $\tilde{\Gamma}$  are strategically equivalent.*

The proof can be found in the full paper (Carminati et al., 2024a).

### 3.3.2 Worst-Case Dimension of the Belief Game

The per-iteration time complexity of **CFR** depends linearly on the size of the game on which the algorithm is applied. Thus, it is critical for complexity analysis to bound the size of the belief game produced by **MakeBeliefGame**.

**Lower Bound.** We first present a lower bound of the worst-case size of the belief game, *i.e.* a worst-case instance of game whose belief game has a large number of histories.

**Theorem 3.14.** *There exists a game  $\Gamma$  with depth  $d$ , information complexity  $k$  and maximum branching factor at a node  $b$  such that the number of nodes in the belief game  $\tilde{\Gamma}$  is  $|\tilde{\mathcal{H}}| \geq b^{2k(d-4)}$ .*

**Upper Bound.** We now present an upper bound on the number of histories of the belief game.

**Theorem 3.15.** *Let  $\Gamma$  be a game with depth  $d$ , information complexity  $k$  and maximum branching factor at a node  $b$ . The number of nodes in the belief game  $\tilde{\Gamma}$  is  $|\mathcal{H}| \leq b^{2kd+d}$ .*

**Discussion.** The bounds presented in this section highlight the main computational limitation of Algorithm `MakeBeliefGame`, the explicit dependence on depth introduced by explicitly using sequences to distinguish information sets in the belief game.

We remark that we can replace  $k$  here with the maximum number of infosets (not the last-infosets) in any public state. We opted not to introduce two different notions of information complexity to have bounds comparable with the TB-DAG ones in Section 3.5.1. We will explore the effects of introducing the different definitions of  $k$  in Section 3.7.3.

### 3.3.3 Regret Minimization on Team Games

This section shows how to find a mixed Nash equilibrium in a generic two-player zero-sum game with imperfect recall  $\Gamma$  by applying `CFR` on the belief game  $\tilde{\Gamma}$  obtained by running Algorithm `MakeBeliefGame` on  $\Gamma$ .

Let  $\tilde{\mathcal{X}}$  and  $\tilde{\mathcal{Y}}$  be the realization-form *mixed* strategy spaces for  $\blacktriangle$  and  $\blacktriangledown$  in  $\tilde{\Gamma}$  derived from the sequence-form representation as in Section 2.2.3. Specifically, vectors  $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}$  are indexed by terminal sequences for  $\blacktriangle$  in  $\tilde{\Gamma}$  (similarly for  $\blacktriangledown$ ). Such a sequence  $\sigma$  can be identified by a list of beliefs and prescriptions, ending in a singleton belief  $\{z\}$  for terminal node  $z \in \mathcal{Z}$ . For any terminal node  $z$ , let  $\Sigma_{\blacktriangle}^z$  be the set of terminal sequences for  $\blacktriangle$  that end at belief  $\{z\}$ . Then computing a Nash equilibrium in  $\tilde{\Gamma}$  (and hence a mixed Nash in  $\Gamma$ ) can be done by solving the max-min problem

$$\max_{\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}} \min_{\tilde{\mathbf{y}} \in \tilde{\mathcal{Y}}} \sum_{z \in \mathcal{Z}} u(z) \sum_{\tilde{\sigma}_{\blacktriangle} \in \Sigma_{\blacktriangle}^z} \tilde{\mathbf{x}}[\tilde{\sigma}_{\blacktriangle}] \sum_{\tilde{\sigma}_{\blacktriangledown} \in \Sigma_{\blacktriangledown}^z} \tilde{\mathbf{y}}[\tilde{\sigma}_{\blacktriangledown}]. \quad (2)$$

This is equivalent to the max-min problem for the coordinator game by setting  $\mathbf{x}[z] := \sum_{\tilde{\sigma}_{\blacktriangle} \in \Sigma_{\blacktriangle}^z} \tilde{\mathbf{x}}[\tilde{\sigma}_{\blacktriangle}]$  (and similar for  $\mathbf{y}$ ). That is, from an optimization perspective, what has happened is that we have constructed sets  $\tilde{\mathcal{X}}$  and  $\tilde{\mathcal{Y}}$  that are described by linear constraints, just like the sequence form, and *project* onto  $\mathcal{X}$  and  $\mathcal{Y}$  respectively, allowing the reformulation and equivalence of problems.

We now analyze the time complexity and regret of running `CFR` on  $\tilde{\Gamma}$ . Fix a player, say,  $\blacktriangle$ . (The same analysis will apply to  $\blacktriangledown$ .) First, recall from Section 2.2.3 that, in a decision problem, a set  $P$  of  $\blacktriangle$ -decision points is called *playable* if there exists a pure strategy of  $\blacktriangle$  that plays to all the decision points in  $P$ . But the size of any playable set  $P$  of  $\blacktriangle$  is at most  $|\mathcal{H}|$ . Further, the branching factor of  $\tilde{\Gamma}$  is at most  $b^k$ , where  $b$  is the branching factor of  $\Gamma$  and  $k$  is the information complexity (see Section 3.3.2). Thus, applying multiplicative weights (MWU) as the local regret minimizer at each decision point and using Proposition 2.3, we have:

**Theorem 3.16.** *After  $T$  iterations of `CFR` on  $\tilde{\Gamma}$  with MWU as the local regret minimizer, the average strategy profile  $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$  is an  $O(\epsilon)$ -Nash equilibrium of  $\Gamma$ , where*

$$\epsilon = |\mathcal{H}| \sqrt{\frac{k \log b}{T}}.$$

*The per-iteration complexity is linear in the size of  $\tilde{\Gamma}$ .*

While the regret above is polynomial in  $\mathcal{H}$ , the per-iteration complexity depends on the size of  $\tilde{\Gamma}$ , which is worst-case exponentially larger than  $\Gamma$ , as shown in Section 3.3.2.

---

**Algorithm DAG-Generic:** Generic construction of a regret minimizer  $\mathcal{R}$  on  $\mathcal{Q}$  from a regret minimizer  $\hat{\mathcal{R}}$  on its tree form  $\hat{\mathcal{Q}}$ .

---

```

1: procedure NEXTSTRATEGY
2:    $\hat{\mathbf{x}}^t \leftarrow \hat{\mathcal{R}}.\text{NEXTSTRATEGY}()$ 
3:   return  $D\hat{\mathbf{x}}^t$ 
4: procedure OBSERVEUTILITY( $\mathbf{u}^t$ )
5:    $\hat{\mathcal{R}}.\text{OBSERVEUTILITY}(D^\top \mathbf{u}^t)$ 

```

---

**Algorithm DAG-CFR:** Counterfactual regret minimization on DAG-form decision problems  $\mathcal{Q}$ . For each decision point  $j$ ,  $\mathcal{R}_j$  is a regret minimizer on  $\Delta(\mathcal{A}(j))$ .

---

```

1: procedure NEXTSTRATEGY
2:    $\mathbf{x}^t[\emptyset] \leftarrow 1$ 
3:   for each decision point  $j$ , in top-down order do
4:      $\mathbf{r}_j^t \leftarrow \mathcal{R}_j.\text{NEXTSTRATEGY}()$ 
5:      $\mathbf{x}^t[j^*] \leftarrow \sum_{p \in P_j} \mathbf{x}^t[p] \mathbf{r}_j^t$ 
6:   return  $\mathbf{x}^t$ 
7: procedure OBSERVEUTILITY( $\mathbf{u}^t$ )
8:    $\mathbf{v}^t \leftarrow \mathbf{u}^t$ 
9:   for each decision point  $j$ , in bottom-up order do
10:     $\mathcal{R}_j.\text{OBSERVEUTILITY}(\mathbf{v}^t[j^*])$ 
11:    for  $p \in P_j$  do  $\mathbf{v}^t[p] \leftarrow \mathbf{v}^t[p] + \langle \mathbf{r}_j^t, \mathbf{v}^t[j^*] \rangle$ 
12:    $t \leftarrow t + 1$ 

```

---

### 3.4 DAG Decision Problems

In this section, we will develop a general theory of DAG-form decision problems, and regret minimization on them, analogous to the tree-form theory in Section 2.2.3. Although our main interest in DAG-form decision-making is its application to two-player imperfect-recall games (which we will develop in Section 3.5), the observations made in this section also have general applicability beyond this setting. For example, since the publications of earlier versions of the present paper, DAG-form decision-making has been applied toward the efficient computation of many other solution concepts, including *linear correlated equilibria* and *optimal extensive-form correlated equilibria* (Zhang and Sandholm, 2022a; Zhang et al., 2022b, 2023a, 2024a,d).

As one may expect, DAG-form decision problems are identical to tree-form decision problems except that the graph of nodes is allowed to be a DAG, albeit with some restrictions.

**Definition 3.17.** A *DAG-form decision problem* is a DAG with a unique source (root node)  $\emptyset$ , wherein each node is either a decision point ( $j \in \mathcal{J}$ ) or an observation point ( $s \in \mathcal{S}$ )<sup>21</sup>, with the following properties:

1. Observation points other than the root have exactly one incoming edge.
2. For any two paths  $p_1$  and  $p_2$  from the root that end at the same node, the last node in common between  $p_1$  and  $p_2$  is a decision point.

As with tree-form decision problems, we will also assume (WLOG) that decision and observation points alternate along every path, and that both the root node and all terminal nodes are observation points. A *pure strategy* is once again an assignment of one action to each decision point. The *DAG form* of a pure strategy is the vector  $\mathbf{x} \in \{0, 1\}^{\mathcal{S}}$ , where  $\mathbf{x}[s] = 1$  if there is *some*  $\emptyset \rightarrow s$  path along which the player plays all actions. A mixed strategy  $\mathbf{x} \in \mathcal{Q}$  is a convex combination of pure strategies. Since decision points can now have multiple parents, we will use  $P_j$  to denote the set of parents of a decision point  $j$ .

---

<sup>21</sup>For most of this thesis, observation points are denoted  $\Sigma$ ; however, here we will need to distinguish between observation points  $s \in \mathcal{S}$  and sequences in the original game. We hence choose different notation.



Like tree-form decision problems, the mixed strategy set in a DAG-form decision problem has a convenient representation using linear constraints, namely:

$$\begin{cases} \mathbf{x}[\emptyset] = 1 \\ \sum_{p \in P_j} \mathbf{x}[p] = \sum_{a \in \mathcal{A}(j)} \mathbf{x}[ja] \quad \text{for all } j \in \mathcal{J}. \end{cases} \quad (3)$$

DAG-form decision problems and tree-form decision problems are closely related. Of course, all tree-form decision problems are DAG-form decision problems. Conversely, any DAG-form decision problem can be thought of as a “compressed” representation of the tree-form decision problem created by separating out all the different paths through the DAG. While this tree will generally be exponentially larger than the DAG, we will find it useful to compare the DAG and tree representations.

We now formulate a general theory of regret minimization for DAG-form decision problems. We will use hats  $(\hat{\mathcal{Q}}, \hat{\mathcal{J}}, \hat{\mathcal{S}}, \hat{\mathbf{x}})$  to denote components of the tree form of a generic DAG-form regret minimizer. For each tree-form observation point  $s \in \hat{\mathcal{S}}$  let  $\delta(s) \in \mathcal{S}$  be the corresponding observation point in  $\mathcal{S}$ . Note that, by construction,  $\delta$  is surjective but not injective unless the DAG happens to be a tree.

We now show how tree-form strategies and utilities correspond to DAG-form strategies and utilities. Concretely, we define a matrix  $\mathbf{D} \in \mathbb{R}^{\mathcal{S} \times \hat{\mathcal{S}}}$  by  $\mathbf{D}\hat{\mathbf{x}}[s] = \sum_{\hat{s}: \delta(\hat{s})=s} \hat{\mathbf{x}}[\hat{s}]$  for all  $\hat{\mathbf{x}} \in \mathbb{R}^{\hat{\mathcal{S}}}$ . This is the matrix of the linear map that transforms tree-form strategies to their corresponding DAG-form strategies. That is,  $\mathbf{D} : \hat{\mathcal{X}} \rightarrow \mathcal{X}$  is a bijection.

Dually, for DAG-form utility vectors  $\mathbf{u} \in \mathbb{R}^{\mathcal{S}}$ , the vector  $\mathbf{D}^\top \mathbf{u} \in \mathbb{R}^{\hat{\mathcal{S}}}$  is a utility vector on the tree form, with the property that  $\langle \mathbf{D}^\top \mathbf{u}, \hat{\mathbf{x}} \rangle = \langle \mathbf{u}, \mathbf{D}\hat{\mathbf{x}} \rangle$  by definition of the inner product. That is, the DAG-form strategy  $\mathbf{D}\hat{\mathbf{x}}$  achieves the same utility against DAG-form utility vector  $\mathbf{u}$  as the tree-form strategy  $\hat{\mathbf{x}}$  achieves against the utility  $\mathbf{D}^\top \hat{\mathbf{u}}$ .

The relationship between trees and DAGs allows us to use *any* regret minimizer on  $\hat{\mathcal{Q}}$  to construct a regret minimizer with the same guarantee on  $\mathcal{Q}$ . We do this in Algorithm **DAG-Generic**.

**Proposition 3.18.** *Let  $\mathcal{R}$  and  $\hat{\mathcal{R}}$  be as in Algorithm **DAG-Generic**. Then the regret of  $\mathcal{R}$  with utility sequence  $\mathbf{u}^1, \dots, \mathbf{u}^T$  is equal to the regret of  $\hat{\mathcal{R}}$  with utility sequence  $\mathbf{D}^\top \mathbf{u}^1, \dots, \mathbf{D}^\top \mathbf{u}^T$ .*

Applying the transformation **DAG-Generic** with **CFR** as the tree-form regret minimizer  $\hat{\mathcal{R}}$ , we arrive at a DAG form of CFR, which can be simulated efficiently: Algorithm **DAG-CFR**. One can think of **DAG-CFR** as a *more efficient implementation* of **CFR** when the decision tree happens to have a DAG structure. Of course, the regret bound  $O(|\mathcal{S}|\sqrt{T})$  is only a worst-case bound; in special cases (such as Theorem 3.16), **CFR** does much better than its worst case, and therefore so will **DAG-CFR**.

Call a utility vector  $\hat{\mathbf{u}}$  *consistent* if it is in the image of  $\mathbf{D}^\top$ . That is (expanding the definition of  $\mathbf{D}^\top$ ),  $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{\mathcal{O}}}$  is consistent if  $\hat{\mathbf{u}}[\hat{s}] = \hat{\mathbf{u}}[\hat{s}']$  if  $\delta(\hat{s}) = \delta(\hat{s}')$ . In essence, a DAG-form regret minimizer is able to “simulate” a tree-form regret minimizer so long as the tree-form regret minimizer’s utilities are always consistent. We now formalize this idea.

**Theorem 3.19** (DAG regret minimization via CFR). **DAG-CFR** produces the same iterates as **DAG-Generic** with **CFR** as  $\hat{\mathcal{R}}$ . Therefore, in particular, the regret of **DAG-CFR** with utility sequence  $\mathbf{u}^1, \dots, \mathbf{u}^T$  is the same as that of **CFR** on the tree form with utility sequence  $\hat{\mathbf{u}}^1 := \mathbf{D}^\top \mathbf{u}^1, \dots, \hat{\mathbf{u}}^t := \mathbf{D}^\top \mathbf{u}^t$ . Moreover, the per-iteration runtime of **DAG-CFR** is linear in the number of edges in the DAG. In particular, taking any reasonably efficient regret minimizer over simplices, the regret of **DAG-CFR** after  $T$  iterations is at most  $O(|\mathcal{S}|\sqrt{T})$ .

---

**Algorithm ConstructTB-DAG:** Constructing the TB-DAG. Inputs: imperfect-recall game  $\Gamma$ , player  $i$

---

```

1: procedure MAKEDECISIONPOINT( $B$ )      ▷  $B \subseteq \mathcal{H}$  is a belief
2:   if a decision point  $j$  with belief  $B$  already exists then return  $j$ 
3:   if  $B = \{z\}$  for  $z \in \mathcal{Z}$  then return new terminal node with belief  $\{z\}$ 
4:    $j \leftarrow$  new decision point with belief  $B$ 
5:   for each prescription  $\mathbf{a} \in \mathcal{A}_i(B)$  do
6:     add edge  $j \rightarrow$  MAKEOBSERVATIONPOINT( $B\mathbf{a}$ )
7:   return  $j$ 
8: procedure MAKEOBSERVATIONPOINT( $H$ )
9:    $s \leftarrow$  new observation point
10:  for each  $B \in \text{SPLITBELIEF}_i(H)$  do
11:    add edge  $s \rightarrow$  MAKEDECISIONPOINT( $B$ )
12:  return  $s$ 

```

---

### 3.5 DAG Decision Problems in Team Games

In Section 3.3.3, it emerged that applying the CFR procedure to the belief game produced by `MakeBeliefGame` suffers from the size of the game to solve, which may grow exponentially fast as shown in Section 3.3.2. In this section, we show how DAG decision problems can greatly reduce the inefficiencies caused by the previous construction.

The main observation is that `MakeBeliefGame` enforces perfect recallness of the belief game by including the players' sequences in the info set definition. On the other hand, the strategic aspect of the game is governed solely by the nodes contained in beliefs. Once the set of possible nodes is fixed, the exact sequence of prescriptions and observations is not relevant, as the game will evolve identically from that point onwards. This observation leads to considering a DAG structure for the decision problems, where decision nodes are identified by beliefs.

The Nash equilibrium problem in  $\tilde{\Gamma}$ , namely (2), indeed guarantees that both players' utility vectors will be consistent with respect to these DAG-form decision problems. We will call the resulting DAG decision problems the *team belief DAGs* (TB-DAGs)<sup>22</sup>. Therefore, using `DAG-CFR` as the regret minimizer for both players, we recover the regret guarantee of Theorem 3.16 with *per-iteration complexity* proportional to the total size of both DAGs.

However, this proposed algorithm still depends on the size of  $\tilde{\Gamma}$ , because, naively, to construct the DAG representations, one first constructs the augmented game  $\tilde{\Gamma}$ , and only then does the merging of decision points to create the DAGs. We therefore describe an algorithm `ConstructTB-DAG` that recursively constructs the team belief DAGs *directly from the original game*  $\Gamma$ , thus bypassing the construction of  $\tilde{\Gamma}$ . Therefore, we have the following result. For each player  $i \in \{\blacktriangle, \blacktriangledown\}$ , let  $E_i$  be the number of edges in the TB-DAG of player  $i$ .

**Theorem 3.20** (TB-DAG and CFR). *Suppose that both players run the algorithm `ConstructTB-DAG` to construct their strategy spaces  $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ , and then run `DAG-CFR`. Then their average strategy profile converges at the rate shown in Theorem 3.16, and the per-iteration runtime complexity is  $O(E_{\blacktriangle} + E_{\blacktriangledown})$ .*

<sup>22</sup>We keep the name *team belief DAG* for continuity with previous versions of the work, even though it applies equally well in the team and imperfect recall settings.

### 3.5.1 Size Analysis of the TB-DAG

The per-iteration runtime above depends on the number of edges in the TB-DAGs, so it is important to bound this number. We will do so now. Here, we use the same notation as in Section 3.3.2.

**Theorem 3.21.** *For each player  $i$ , we have  $E_i \leq |\mathcal{H}|(b+1)^{k+1}$ .*

Thus, from Theorem 3.20 and Theorem 3.21, it follows that:

**Theorem 3.22** (Main theorem). *Any given imperfect-recall game  $\Gamma$  can be solved by constructing the TB-DAGs using **ConstructTB-DAG** and running **DAG-CFR**. After  $T$  iterations, the average strategy profile will be an  $O(\epsilon)$ -Nash equilibrium where  $\epsilon$  is as in Theorem 3.16. The per-iteration complexity is  $O(|\mathcal{H}|(b+1)^{k+1})$ .*

Before proceeding, it is instructive to briefly compare Theorem 3.22 to Theorem 3.15. The latter result gives a per-iteration complexity that is  $O(b^{d(2k+1)})$ . Thus, Theorem 3.22 is strictly superior: for Theorem 3.15 to be superior, we would need to have  $b^{d(2k+1)} < |\mathcal{H}|(b+1)^{k+1}$ , which is impossible for  $d \geq 1, k \geq 1, b \geq 2$ . We give a more detailed comparison between the two bounds in Section 3.7.3.

### 3.5.2 Fixed-Parameter Hardness

Given the above result, one may ask whether the  $b$  can be removed more generally. It turns out that it cannot. Before proceeding, we need to introduce some basic concepts surrounding *fixed-parameter tractability*.

**Definition 3.23.** A problem is *fixed-parameter tractable* with respect to a parameter  $k$  if it admits an algorithm whose runtime on inputs of length  $N$  is  $f(k)\text{poly}(N)$ , for some arbitrary function  $f$ .

The  $k$ -*CLIQUE* problem is to, given a graph  $\Gamma$  and an integer  $k$ , decide where  $\Gamma$  has a  $k$ -clique. The computational assumption  $\text{FPT} \neq \text{W}[1]$  states that  $k$ -*CLIQUE* is not fixed-parameter tractable. It is implied by the exponential time hypothesis (Chen et al., 2005).

**Theorem 3.24.** *Assuming  $\text{FPT} \neq \text{W}[1]$ , there is no algorithm for computing the mixed Nash value of a one-player game of imperfect recall whose runtime has the form  $f(k)\text{poly}(|\mathcal{H}|)$  where  $f$  is an arbitrary function.*

Thus, it is impossible to replace the  $b$  in Theorem 3.22 with any absolute constant.

### 3.5.3 Branching Factor Reduction

Despite the worst-case hardness of removing the  $b$  in Theorem 3.22, it turns out that, for a natural class of games, we *can* remove  $b$ . In this subsection we will discuss games with *action recall*, and prove that in such games, it is without loss of generality to assume that the branching factor is 2. Intuitively, a player  $i$  has action recall if it remembers the full sequence of actions she has taken in the past (including the timesteps at which such actions were taken). More formally:

**Definition 3.25.** At a node  $h \in \mathcal{H}$ , let  $(a_1, \dots, a_L) \in \mathcal{A}^L$  be the list of actions taken on the  $\emptyset \rightarrow h$  path. Define the *action sequence* of player  $i$  as the sequence  $(a'_1, \dots, a'_L) \in (\mathcal{A} \sqcup \{\perp\})^L$  where  $a'_\ell = a_\ell$  if action  $a_\ell$  was taken by player  $i$ , and  $a'_\ell = \perp$  otherwise. We say that player  $i$  has *action recall* if, for every infoset  $I$  of player  $i$ , every node in  $I$  shares the same action sequence.

**Theorem 3.26.** *Given a two-player imperfect-recall game  $\Gamma$  where both players have perfect action recall, there exists another strategically-equivalent game  $\Gamma'$  such that the branching factor of  $\Gamma'$  is at most 2 at each  $h \in \mathcal{H}_\blacktriangle \cup \mathcal{H}_\blacktriangledown$ , the parameter  $k$  in  $\Gamma'$  is the same as it in  $\Gamma$ , and the size of the game has increased by a factor of  $O(\log |\mathcal{A}|)$ .*

	Team vs Player	Team vs Team
TMECor	NP-complete (Koller and Megiddo, 1992)	$\Delta_2^P$ -complete (This paper, Theorems 3.31 and 3.32)
TME	NP-complete (Koller and Megiddo, 1992)	$\Sigma_2^P$ -complete (This paper, Theorems 3.29 and 3.30)

**Table 4:** Summary of most of the complexity results shown in Section 3.6.

**Corollary 3.27.** *In games where both players have action recall, Theorem 3.22 applies with the per-iteration runtime replaced with  $O(3^k |\mathcal{H}| \log |\mathcal{A}|)$ .*

### 3.6 Complexity of Adversarial Team Games

Here, we state and prove several results about the *complexity* of finding various equilibria in timeable two-player zero-sum games of imperfect recall.

In all cases, the goal is to solve the following promise problem: given game  $\Gamma$ , threshold value  $v$ , and error  $\epsilon > 0$  (where all the numbers are rational), determine whether the (mixed or behavioral) value of the game is  $\geq v$ , or  $< v - \epsilon$ . The allowance of an exponentially-small error is to circumvent issues of bit complexity that arise due to the fact that exact behavioral max-min strategies may not have rational coefficients (Koller and Megiddo, 1992). Throughout this section, it will often be convenient to formulate the hardness gadgets in terms of adversarial team games. We will thus freely utilize the analogy between adversarial team games and coordinator games. For mixed Nash and behavioral Nash respectively, we will refer to the problems as MIXED and BEHAVIORAL.

Although we do not explicitly state it in the theorem statements, all the hardness results are proven by constructing adversarial team games in which both teams have a constant number of players.

**Theorem 3.28** (Koller and Megiddo, 1992; Chu and Halpern, 2001; von Stengel and Forges, 2008). *Finding the optimal strategy in a one-player, timeable game of imperfect recall is NP-hard.*

The above result also shows, by the PCP theorem (Håstad, 2001), that there exists an absolute constant  $\epsilon$  such that computing the optimal value in a team game with no adversary to accuracy  $\epsilon$  is NP-hard. Finally, the information complexity of the game used in the above construction is<sup>23</sup>  $k = n$ , and the branching factor can be made an absolute constant by splitting the root chance node into  $\Theta(\log m)$  layers. Finally, the size of the game is  $O(mn)$ . Thus, Theorem 3.22 implies a SAT-solving algorithm whose runtime is  $2^{O(n)}$ . Thus, in particular, the appearance of  $k$  in the exponent in Theorem 3.22 is unavoidable: if the  $k$  were replaced by any  $o(k)$  term, then SAT would have an  $2^{o(n)}$ -time algorithm, violating the commonly-believed exponential time hypothesis.

<sup>23</sup>Here we use the ordering of the players: namely, we have  $k = n$  only because P1 plays before P2. If the order of the players were flipped, we would instead have  $k = m$ .

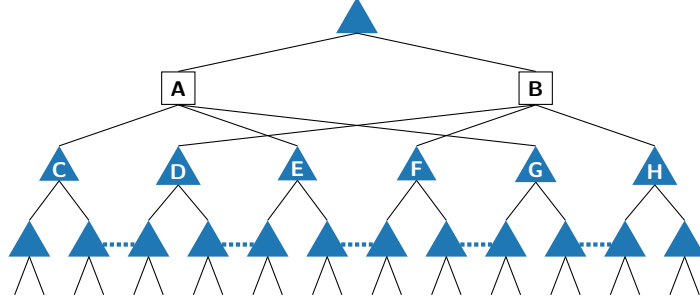


Figure 5: A game showing that public state-based approaches do not subsume inflation.

### 3.6.1 Behavioral Max-Min Strategies

We first show results for BEHAVIORALMAXMIN. In particular, we will show that it is  $\Sigma_2^P$ -complete, first by showing inclusion and then constructing a gadget game to show completeness. (Recall that the inclusion will require an  $\epsilon$ -approximation because exact behavioral max-min strategies may contain irrational values.)

**Theorem 3.29.** BEHAVIORALMAXMIN is in  $\Sigma_2^P$ . If  $\blacktriangledown$  has perfect recall, it is in NP.

**Theorem 3.30.** BEHAVIORALMAXMIN is  $\Sigma_2^P$ -hard, even for team games with a constant number of players and no chance.

### 3.6.2 Mixed Nash Equilibria

We now show results for MIXEDNASH, namely, we will show that MIXEDNASH is  $\Delta_2^P$ -complete, again by showing inclusion first and then completeness. Unlike for BEHAVIORALMAXMIN (Theorem 3.29), here we will directly construct a separation oracle, and thus be able to recover algorithms for *exact* computation.

**Theorem 3.31.** MIXEDNASH is in  $\Delta_2^P$ , even for exact computation ( $\epsilon = 0$ ).

**Theorem 3.32.** MIXEDNASH is  $\Delta_2^P$ -hard, even for team games with a constant number of players and no chance.

## 3.7 Discussion

In the following section we discuss important details that may help the interested reader in clarifying some technical aspects of our contributions.

### 3.7.1 Public States vs Observations

In this section, we discuss in depth the difference between public *states* and public *observations*. Intuitively, the difference is that observations are *localized* to a particular node in the TB-DAG: if a fact is public to the team *conditional on the part of the team strategy that has been played to reach this point*, then it is an observation. On the other hand, public *states* only encode *unconditionally* public information. As we will see, using observations is strictly preferable to public states from both conceptual and theoretical perspectives.

**Comparison to using public states.** We envision an alternative construction of the TB-DAG in which the team coordinator observes only the *public state* containing the current node. That is, the definition of SPLITBELIEF is replaced by:

$$\text{SPLITBELIEF}_i^{\text{pub}}(H, h) := H \cap P \text{ where } h \in P \in \mathcal{P}_i.$$

and  $\text{SPLITBELIEF}_i^{\text{pub}}(H)$  defined analogously. Then, in **ConstructTB-DAG**, we replace  $\text{SPLITBELIEF}_i(H)$  with  $\text{SPLITBELIEF}_i^{\text{pub}}(H)$ . We will call this new construction the *public-state TB-DAG* and spend the rest of this subsection contrasting it with the (observation) TB-DAG constructed by **ConstructTB-DAG**.

Our first result is that the TB-DAG can never be too much larger than the public state TB-DAG:

**Proposition 3.33.** *Let  $N$  and  $N'$  be the number of nodes in the TB-DAG and public state TB-DAG respectively. Then  $N \leq 2pN'$ , where  $p$  is the largest size (in number of nodes) of any belief in the public state TB-DAG.*

Thus, using observations is never much worse than using public states.

**Comparison to using inflated public states.** Previous works (Zhang and Sandholm, 2022b; Carminati et al., 2022) used public states and required to *inflate* the information partition of the team before the new representation can be constructed. *Complete inflation* (Kaneko and Kline, 1995), which we simply call *inflation* for short, is an algorithm that splits an info set  $I$  into two info sets  $I = I_1 \sqcup I_2$  if no pure strategy of the team can simultaneously play to a node in  $I_1$  and a node in  $I_2$ , and repeats this process until no more such splits are possible. This preserves strategic equivalence. However, inflation can lead to the break-up of public states, which, in turn, reduces the size of public state TB-DAG.

Indeed, consider the game in Figure 5. Due to the information sets marked in the last layer of the game tree, the connectivity graph contains a path C—D—E—...—H. Therefore, {C, D, ..., H} form a public state. Also, it is possible for the combinations CEG and DFH to be reached (if the player at the root plays left or right, respectively). Therefore, CEG and DFH are beliefs in the public-state TB-DAG. In the observation TB-DAG, consider, for example, what happens if the left action is played at the root so that C, E, and G are all reached. Note that there are no edges connecting C, E, and G—the path connecting C to E in the connectivity graph passes through D, which is not reached; therefore, C, E, and G are three different observations and hence three different beliefs, resulting in an exponentially-smaller TB-DAG. Inflation would remove the nontrivial information sets in the second black layer, which would ultimately have the same effect in this example as using observations.

The number 3 is not special in this construction; it can be increased arbitrarily by simply increasing the number of children of A and B. Therefore, in particular, one can construct a family of games in which the public state TB-DAG (without inflation) has exponential size, while the (observation) TB-DAG has polynomial size. This is why Zhang and Sandholm (2022b) and Carminati et al. (2022) insist that inflation be done as a preprocessing step before beginning their constructions.

The use of observations, however, removes the need for this step:

**Proposition 3.34.** *Given any team decision problem  $\mathcal{T}$ , the TB-DAG of  $\mathcal{T}$  is the same no matter whether inflation is applied to  $\mathcal{T}$  before the construction.*

Although inflation *can* be performed efficiently, not requiring it as a preprocessing step simplifies the code and makes for a conceptually cleaner construction. However, the benefits of observations go beyond making inflation unnecessary. In fact, even with inflation, there are still cases in which using observations instead represents an exponential improvement.

**Proposition 3.35.** *There exists a family of team decision problems in which the TB-DAG has polynomial size, but the public state TB-DAG has exponential size, even if inflation is applied as a preprocessing step before building the latter.*

A practical experiment backs up these results. When  $C = 16$ , using observations generates a DAG with around 1000 edges; using public states generates a DAG with 30 million edges.

### 3.7.2 Tree vs DAG Representation

Here we give an explicit example in which the TB-DAGs will be exponentially smaller than the game tree generated by `MakeBeliefGame`. This construction would work for most nontrivial adversarial team games, but for concreteness, consider the game  $\Gamma$  depicted in Figure 1. Call the leftmost terminal node in that diagram  $z$ . Consider adding another copy of  $\Gamma$  rooted at node  $z$ , and then repeating this process until  $\ell$  copies of the game tree have been created, thus forming a game  $\Gamma^\ell$ . That is,  $\Gamma^\ell$  is the game in which  $\Gamma$  is played repeatedly until  $\ell$  repetitions have been reached, or the terminal node reached is not  $z$ .

Note that, when running `MakeBeliefGame` on  $\Gamma$ , multiple copies of node  $z$  will appear. Thus, the number of nodes in the auxiliary game will be exponential in  $\ell$ . However, in the TB-DAG, after the  $i$ th repetition of the game finishes, the belief will always be  $\{z_i\}$  (where  $z_i$  is the copy of  $z$  in the  $i$ th repetition of the game). Thus, the size of the TB-DAG will scale linearly with  $\ell$ . Thus, as  $\ell$  grows, the TB-DAG will be exponentially smaller than the auxiliary game, and in particular the TB-DAG will have polynomial size while the auxiliary game will have exponential size.

### 3.7.3 Definition of Information Complexity and Comparison of Bounds

We discuss the comparison between the bounds from Theorem 3.15 and Theorem 3.21 in more detail.

In Section 3.3, we defined the information complexity as the maximum number of *last-infosets* in any public state. This definition was made with Theorem 3.22 in mind, because it is the correct parameterization for that result. For Theorem 3.15, however, we could have used a tighter parameterization. In particular, we could have defined a parameter  $\kappa$  as the number of infosets (not last-infosets) in any public state. Then  $O(b^{2\kappa d+d})$  would be a valid upper bound in Theorem 3.15. One might ask how this new upper bound compares to that of Theorem 3.22. To this end, we now compare the two bounds.

**Lemma 3.36.**  $k \leq \kappa d$ .

Thus, the bound in Theorem 3.22 is at most

$$|\mathcal{H}|(b+1)^{k+1} \leq |\mathcal{H}|(b+1)^{\kappa d} < |\mathcal{H}|b^{2\kappa d} \leq b^{2\kappa d+d}$$

where we use the bounds  $b \geq 2$  (which holds for every nontrivial game) and  $|\mathcal{H}| \leq b^d$ . Thus, we conclude that the bound in Theorem 3.22 is always strictly tighter than the bound in Theorem 3.15.

We also remark that in any case  $\kappa \leq |\mathcal{H}|$  is a loose bound that still ensures that the overall bound in Theorem 3.16 is polynomial in  $|\mathcal{H}|$ .

### 3.7.4 Connection with Tree Decomposition

The public *state* TB-DAG can be viewed from the perspective of graphical models, specifically, using *tree decompositions*. Here, we review tree decompositions and show the tree decomposition-based perspective of the public-state TB-DAG.

**Definition 3.37.** Given a (simple) graph  $\mathcal{G} = (V, E)$ , a *tree decomposition*<sup>24</sup> is a tree  $\mathcal{J}$ , with the following properties:

1. the nodes of  $\mathcal{J}$  are subsets of  $V$ , called *bags*;
2. for each edge  $(u, v) \in E$ , there is a bag containing both  $u$  and  $v$ ; and
3. for each vertex  $u \in V$ , the subset of nodes of  $\mathcal{J}$  whose bags contain  $u$  is connected.

Consider an arbitrary set of the form

$$\mathcal{X} = \{\mathbf{x} \in \{0, 1\}^n : g_k(\mathbf{x}) = 0 \quad \forall k \in [m]\}$$

where the  $g_k$ s are arbitrary constraints. Each constraint  $g_k$  has a *scope*  $S_k \subseteq [n]$  of variables on which it depends. The *dependency graph* of  $\mathcal{X}$  is the graph  $\mathcal{G}_{\mathcal{X}}$  whose nodes are the integers  $1, \dots, n$ , and where there is an edge  $(i, j)$  if there is a constraint whose scope  $S_k$  contains both  $i$  and  $j$ . For a subset  $U \subseteq [n]$ , a vector  $\tilde{\mathbf{x}} \in \{0, 1\}^U$  is *locally feasible* if  $\tilde{\mathbf{x}} = \mathbf{x}_U$  for some  $\mathbf{x} \in \mathcal{X}$ . We will use  $\mathcal{X}_U$  to denote the set of all locally feasible vectors on  $U$ . Of course,  $\mathcal{X}_{[n]} = \mathcal{X}$ .

The main result of interest to us is a corollary of the junction tree theorem (e.g., (Wainwright and Jordan, 2008)), which allows an arbitrary set  $\text{co } \mathcal{X}$  to be described with a constraint system whose size is related to the sizes of tree decompositions of  $\mathcal{G}_{\mathcal{X}}$ .

**Theorem 3.38** (Wainwright and Jordan, 2008). *Let  $\mathcal{J}$  be a tree decomposition of  $\mathcal{G}_{\mathcal{X}}$ . Then  $\mathbf{x} \in \mathcal{X}$  if and only if there are vectors  $\lambda_U \in \Delta(\mathcal{X}_U)$  for each bag  $U$  of  $\mathcal{J}$ , such that:*

$$\begin{aligned} \mathbf{x}_U &= \sum_{\tilde{\mathbf{x}} \in \mathcal{X}_U} \lambda_U[\tilde{\mathbf{x}}] \cdot \tilde{\mathbf{x}} && \forall \text{ bags } U \text{ in } \mathcal{J} \\ \sum_{\substack{\tilde{\mathbf{x}} \in \mathcal{X}_U \\ \tilde{\mathbf{x}}_{U \cap V} = \tilde{\mathbf{x}}^*}} \lambda_U[\tilde{\mathbf{x}}] &= \sum_{\substack{\tilde{\mathbf{x}} \in \mathcal{X}_V \\ \tilde{\mathbf{x}}_{U \cap V} = \tilde{\mathbf{x}}^*}} \lambda_V[\tilde{\mathbf{x}}] && \forall \text{ edges } (U, V) \text{ of } \mathcal{J} \text{ and } \tilde{\mathbf{x}}^* \in \mathcal{X}_{U \cap V} \end{aligned}$$

Intuitively, the first constraint says that every  $\mathbf{x}_U$  must be a convex combination of locally feasible  $\tilde{\mathbf{x}} \in \mathcal{X}_U$ . This is of course a necessary condition. The second constraint says that marginal probabilities on edges  $(U, V)$  must be consistent with each other. This is also clearly a necessary condition, so the difficulty of proving the above result lies in showing that these two constraints are *sufficient*. We will not prove the result here, but we will use it as a black box.

In this section, we will work with a representation slightly different from the realization form. For a player  $i$  in a coordinator game  $\Gamma$  and a pure strategy of that player, the *history form* of the strategy as the vector  $\mathbf{x} \in \{0, 1\}^{\mathcal{H}}$  where  $\mathbf{x}[h] = 1$  if and only if the team plays all actions on the  $\emptyset \rightarrow h$  path. (Of course, the realization form is just the subvector of  $\mathbf{x}$  indexed by  $\mathcal{Z}$ .) As usual we will use  $\mathcal{X}$  for the set of pure strategies in history form, and  $\mathbf{x} = \text{co } \mathcal{X}$ . The history form is the set of vectors  $\mathbf{x} \in \{0, 1\}^{\mathcal{H}}$  satisfying the following constraint system.

$$\begin{aligned} \mathbf{x}[\emptyset] &= 1 \\ \mathbf{x}[ha] &= \mathbf{x}[h] && \text{if } h \notin \mathcal{H}_i \\ \mathbf{x}[h] &= \sum_{a \in \mathcal{A}(h)} \mathbf{x}[ha] && \text{if } h \in \mathcal{H}_i \\ \mathbf{x}[ha]\mathbf{x}[h'] &= \mathbf{x}[h'a]\mathbf{x}[h] && \text{if } h, h' \in I \in \mathcal{I}_i; a \in \mathcal{A}(h) \end{aligned}$$

<sup>24</sup>also known as a *clique tree* or *junction tree*



$\Gamma$	Nodes $ \mathcal{H} $	Original game $\Gamma$				Information		Nodes $ \mathcal{H} $	Belief Game $\bar{\Gamma}$				Team $\blacktriangle$ 's DAG		Team $\blacktriangledown$ 's DAG	
		Infosets $ \mathcal{I}_\blacktriangle $	Infosets $ \mathcal{I}_\blacktriangledown $	Sequences $ \Sigma_\blacktriangle $	Sequences $ \Sigma_\blacktriangledown $	$\max_P  P $	$k$		Infosets $ \bar{\mathcal{I}}_\blacktriangle $	Infosets $ \bar{\mathcal{I}}_\blacktriangledown $	Sequences $ \bar{\Sigma}_\blacktriangle $	Sequences $ \bar{\Sigma}_\blacktriangledown $	Vertices $ \mathcal{V}_\blacktriangle \cup \mathcal{V}_\blacktriangledown $	Edges $ \mathcal{E}_\blacktriangle $	Vertices $ \mathcal{V}_\blacktriangledown \cup \mathcal{V}_\blacktriangle $	Edges $ \mathcal{E}_\blacktriangledown $
${}^3\text{K3} \{3\}$	151	24	12	48	24	6	6	2119	486	12	1062	24	487	918	37	36
${}^3\text{K4} \{3\}$	601	32	16	64	32	12	8	45,049	4487	16	9800	32	2100	6711	49	48
${}^3\text{K6} \{3\}$	3001	48	24	96	48	30	12	6,768,601	267,184	24	574,588	48	54,255	336,944	73	72
${}^3\text{K8} \{3\}$	8401	64	32	128	64	56	16	617,929,873	13,194,749	32	27,978,704	64	1,783,926	15,564,765	97	96
${}^3\text{K12} \{3\}$	33,001	96	48	192	96	132	24	—	—	—	—	—	—	—	—	—
${}^4\text{K5} \{3,4\}$	7801	80	80	160	160	20	10	577,764,601	102,725	10,385	221,810	21,740	26,566	124,875	4621	15,415
${}^4\text{K5} \{4\}$	7801	120	40	240	80	60	15	174,273,721	11,739,640	40	25,581,730	80	998,471	4,658,070	121	120
${}^3\text{L133} \{3\}$	12,688	456	228	912	456	9	6	1,293,658	96,115	228	208,136	456	23,983	49,005	685	684
${}^3\text{L143} \{3\}$	40,409	800	400	1600	800	16	8	52,745,745	2,625,209	400	5,736,592	800	139,964	417,027	1201	1200
${}^3\text{L151} \{3\}$	19,981	1000	500	2000	1000	20	10	152,692,141	16,564,617	500	36,016,124	1000	150,707	496,196	1501	1500
${}^3\text{L153} \{3\}$	98,606	1240	620	2480	1240	25	10	1,833,113,016	67,400,747	500	147,671,104	1240	855,397	3,486,091	1861	1860
${}^3\text{L223} \{3\}$	15,659	1260	630	2884	1442	4	4	521,285	47,579	812	100,420	1624	32,750	45,913	2437	2436
${}^4\text{L523} \{3\}$	1,299,005	99,168	49,584	246,304	123,152	4	4	178,141,285	19,499,329	73,568	40,224,140	147,136	2,911,352	4,183,685	220,705	220,704
${}^4\text{L133} \{3,4\}$	159,001	1632	1632	3264	3264	9	6	985,916,371	475,081	135,322	1,011,500	292,400	79,351	158,058	75,157	155,475
${}^3\text{D3} \{3\}$	27,622	1023	513	2046	1020	9	6	70,704,118	3,235,954	765	5,501,789	1272	91,858	215,967	1522	1521
${}^3\text{D4} \{3\}$	524,225	10,924	5460	21,840	10,920	16	8	—	—	—	—	—	4,043,377	13,749,608	16,381	16,380
${}^4\text{D3} \{2,4\}$	663,472	6144	6144	12,288	12,285	9	6	—	—	—	—	—	514,120	1,217,310	486,442	1,155,144
${}^6\text{D2} \{2,4,6\}$	524,225	4096	4096	8190	8190	8	6	5,879,066,753	1,094,865	701,001	1,869,170	1,202,948	254,758	457,795	218,570	389,995
${}^6\text{D2} \{4,6\}$	524,225	5704	2488	10,920	5460	16	8	4,992,649,921	15,032,900	33,905	25,363,692	57,194	991,861	2,029,546	46,236	60,717
${}^6\text{D2} \{6\}$	524,225	6584	1608	12,922	3458	32	10	2,126,796,737	126,748,497	2532	208,964,598	4382	3,158,364	7,395,885	5551	5550

**Table 6:** Game sizes of the equivalent representations proposed in the paper (i.e. belief game and TB-DAG) on several standard parametric benchmark team games. See Section 3.8 for a description of the games, and for a detailed description of the meaning of each column. Values denoted with ‘—’ are missing due to out-of-time or out-of-memory errors.

This constraint system defines a dependency graph  $\mathcal{G}_\mathcal{X}$ , whose nodes are nodes of the tree, and in which there is an edge  $(h, h')$  if either  $h'$  is a child of  $h$ , or  $h$  and  $h'$  are in the same infoset of player  $i$ .

Now consider the following tree decomposition of  $\mathcal{J}$  of  $\mathcal{G}_\mathcal{X}$ . For each public state  $P$ , the tree decomposition  $\mathcal{J}$  has a bag  $U_P$  that contains all nodes in  $P$  and all children of nodes in  $P$ . The edges of  $\mathcal{J}$  are the obvious edges, connecting each  $U_P$  to  $U_{P'}$  if  $U_P \cap U_{P'} \neq \emptyset$ .

One can check that, up to trivial reformulations (that is, removal of redundant variables and constraints), the constraint system from Theorem 3.38 associated with  $\mathcal{J}$  is identical to the constraint system associated with the public state TB-DAG (via (3)). Thus, it is possible to interpret the public state TB-DAG entirely from the point of view of tree decompositions. We do not take this perspective in the rest of the paper because using beliefs is more interpretable and understandable from a game-theoretic perspective.

### 3.7.5 Postprocessing Techniques that Can Be Used to Shrink the TB-DAG

In practice, **ConstructTB-DAG** is suboptimal in several ways. Here, we state some straightforward postprocessing techniques that can be used to shrink the size of the TB-DAG. These do not affect the theoretical statements as the primary focus of those is isolating the dependency on our parameters of interest, but they can significantly affect the practical performance, so we apply them in the experiments.

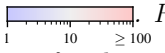
1. If two terminal nodes  $z, z'$  have the same sequence, we remove one of them (say,  $z'$ ) from our DAG because it is redundant, and alias  $\mathbf{x}[\{z'\}]$  to  $\mathbf{x}[\{z\}]$ . If this removal causes a section of the DAG to no longer contain any terminal descendants, we also remove that section.
2. If a decision point in the TB-DAG has (at most) one parent and (at most) one child, we remove the decision point and directly connect the parent observation node to the grandchild decision points.

In particular, if the team has perfect recall, the above two optimizations are sufficient for the TB-DAG to coincide with the sequence form.

## 3.8 Experiments

This section investigates the empirical benefits brought about by applying the TB-DAG when computing mixed-Nash equilibria. As highlighted in Section 3.1, the literature on team games has been the one most concerned with the efficient computation of mixed Nash, with different works establishing benchmarks and proposing algorithms. We will, therefore, focus on comparing our approach against those previous related works. Our main results are reported in Table 6, which reports the size of the original games and our

Game $\{\heartsuit\}$ $\Gamma$	Original game			TB-DAG			EFG		
	$\blacktriangle$ Value	Nodes	Information	This paper (CFR)			ZFCSS22 (CG)		
	$u^*$	$ \mathcal{H} $	$\max_{\mathcal{P}}  P $ $k$	Init	$\epsilon=10^{-3}$	$\epsilon=10^{-4}$	Init	$\epsilon=10^{-3}$	$\epsilon=10^{-4}$
${}^3\text{K3}$ {3}	0.000	151	6 6	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s
${}^3\text{K4}$ {3}	-0.042	601	12 8	0.01s	0.00s	0.00s	0.00s	0.01s	0.02s
${}^3\text{K6}$ {3}	-0.024	3001	30 12	1.03s	0.03s	0.12s	0.00s	0.14s	0.14s
${}^3\text{K8}$ {3}	-0.019	8401	56 16	1m6s	4.73s	32.36s	0.01s	0.23s	0.32s
${}^3\text{K12}$ {3}	-0.014	33,001	132 24	—	oom	oom	0.01s	0.84s	1.39s
${}^4\text{K5}$ {3,4}	-0.037	7801	20 10	0.55s	0.03s	0.05s	—	—	—
${}^4\text{K5}$ {4}	-0.030	7801	60 15	13.71s	1.59s	6.34s	—	—	—
${}^3\text{L133}$ {3}	0.215	12,688	9 6	0.49s	0.02s	0.05s	0.02s	24.89s	45.96s
${}^3\text{L143}$ {3}	0.107	40,409	16 8	1.39s	0.10s	0.48s	0.05s	2m 4s	6m 3s
${}^3\text{L151}$ {3}	-0.019	19,981	20 10	1.54s	0.18s	0.50s	0.04s	3.06s	13.98s
${}^3\text{L153}$ {3}	0.024	98,606	25 10	16.03s	1.24s	4.94s	0.12s	7m 23s	28m 13s
${}^3\text{L223}$ {3}	0.516	15,659	4 4	0.13s	0.03s	0.08s	0.05s	13.48s	18.53s
${}^3\text{L523}$ {3}	0.953	1,299,005	4 4	18.02s	11.26s	24.86s	6.83s	> 6h	> 6h
${}^4\text{L133}$ {3,4}	0.147	159,001	9 6	2.03s	0.21s	0.92s	—	—	—
${}^3\text{D3}$ {3}	0.284	27,622	9 6	0.80s	0.11s	0.40s	0.09s	11.05s	11.05s
${}^3\text{D4}$ {3}	0.284	524,225	16 8	1m3s	22.54s	1m 28s	1.57s	3h 19m	3h 19m
${}^4\text{D3}$ {2,4}	0.200	663,472	9 6	27.05s	2.31s	4.70s	—	—	—
${}^6\text{D2}$ {2,4,6}	0.072	524,225	8 6	10.74s	1.72s	4.26s	—	—	—
${}^6\text{D2}$ {4,6}	0.265	524,225	16 8	16.55s	3.80s	11.09s	—	—	—
${}^6\text{D2}$ {6}	0.333	524,225	32 10	31.00s	30.20s	1m 11s	—	—	—

**Table 7:** Runtime of our CFR-based algorithm (column ‘This paper’) using the team belief DAG form, compared to the prior state-of-the-art algorithms based on linear programming and column generation by Zhang et al. (2022b) (‘ZFCSS22’), on several standard parametric benchmark games. See Section 3.8 for a description of the games. Column ‘Init’ represents the time needed to construct the structures needed for solving the games. This corresponds to fully exploring the TB-DAG and computing its full representation in memory in the TB-DAG case. Missing or unknown values are denoted with ‘—’. For each row, the background color of each runtime column is set proportionally to the ratio with the best runtime for the row, according to the logarithmic color scale . Runtimes that are more than two orders of magnitude larger than the best runtime for the row (i.e., for which  $R > 10^2$ ) are colored as if  $R = 10^2$ .

derived representations, and in Table 7, which reports the time required to solve those instances up to an approximation factor.

### 3.8.1 Experimental Setting

First, we give a complete description of the experimental setting in which the different algorithms are tested.

**Game instances.** We run experiments on commonly adopted parametric benchmarks in the team games literature. The following is the naming convention adopted for the instances considered:

- ${}^n\mathbf{Kr}$ :  $n$ -player Kuhn poker with  $r$  ranks (Kuhn, 1950b).
- ${}^n\mathbf{Lbrs}$ :  $n$ -player Leduc poker with a  $b$ -bet maximum in each betting round,  $r$  ranks, and  $s$  suits (Southey et al., 2005).
- ${}^n\mathbf{Dd}$ :  $n$ -player Liar’s Dice with one  $d$ -sided die for each player (Lisỳ et al., 2015).

The full description of these games can be found in Farina et al. (2021a). For each game, the players belonging to team  $\blacktriangledown$  are represented along with the name. For example,  ${}^4\mathbf{L133}\{3,4\}$  indicates a 4-player Leduc poker game with 1 bet each round, 3 ranks, 3 suits, where players 3 and 4 belong to team  $\blacktriangledown$  and are therefore coordinated by player  $\blacktriangledown$ .

**CFR Variant used.** We implemented the *Predictive CFR+* (*PCFR+*) (Farina et al., 2021c) state-of-the-art variant of CFR on the TB-DAG. *PCFR+* is a predictive regret minimization algorithm and uses quadratic averaging of iterates. At each time  $t$ , we use the previous utility vector for each time as the prediction for the next. We remark that applying the CFR algorithm on the belief game and on the TB-DAG leads to identical iterations since the two representations are structurally equivalent (as proven in Section 3.5), and CFR is a deterministic algorithm. We therefore focus on the TB-DAG representation due to its efficiency. We also remark that the optimizations discussed in Sections 3.5.3 and 3.7.5 are applied during the experiments.

**Baselines.** We use the column generation framework of Farina et al. (2021a) and refined by Zhang et al. (2022b) (henceforth “**ZFCS22**”) as the prior state-of-the-art algorithm to compare the performance of CFR on the team belief DAG. ZFCS22 belongs to the family of column generation approaches adopted in the past literature in team games. ZFCS22 iteratively refines the strategy of each team by solving best-response problems using a tight integer program derived from the theory of extensive-form correlation (von Stengel and Forges, 2008). We used the original code by the authors, which was implemented for three-player games in which a team of two players faces an opponent.

**Hardware used.** All experiments were run on a 64-core AMD EPYC 7282 processor. Each algorithm was allocated a maximum of 4 threads, 60GBs of RAM, and a time limit of 6 hours. ZFCS22 uses the commercial solver Gurobi to solve linear and integer linear programs. All CFR implementations are single-threaded, while we allowed Gurobi to use up to four threads.

### 3.8.2 Discussion of the Results

We now discuss the empirical results obtained by our algorithms.

**Representation vs Game size.** We analyze the size results from Table 6. The different orders of magnitude of the size of each representation and the original game highlight how the belief game construction increases the size of the game. Moreover, the striking difference between the two equivalent approaches of belief game and TB-DAG motivates the introduction of the latter: the direct construction of a decision problem and the more efficient representation brought by the DAG structure allow the construction of a substantially smaller representation. The benefits of the DAG imperfect-recall structure are especially beneficial in the case of Liar’s Dice instances, which have a larger depth of the game tree. Overall, this comparison confirms the results from the worst-case bounds from Sections 3.3.2, 3.5.1 and 3.7.3. The exponential factor of inefficiency between the two representations agrees with the results from the discussion in Section 3.7.2.

There are also some minor remarks that are worth to be made. Whenever  $\blacktriangledown$  is a perfect-recall player (equivalently, when the team  $\blacktriangledown$  is composed of a single player), our constructions never increase the size of its decision problem. In the case of the belief game, we have that the adversary retains an identical number of information sets and sequences. In the case of the TB-DAG, the correspondence is  $|\mathcal{D}_{\blacktriangledown}| = |\tilde{\mathcal{I}}_{\blacktriangledown}| + 1$  and  $|\mathcal{S}_{\blacktriangledown}| = |\tilde{\Sigma}_{\blacktriangledown}|$

**Running time.** We focus on the time performance of CFR applied to the games from Table 7. The main observation is that the TB-DAG approach combined with the CFR algorithm has good performance in most of the games traditionally employed in the team game literature. In particular, impressive performance is achieved in games where the information complexity is low. This is the case of Leduc and Liar’s Dice benchmarks (whose number of infosets and sequences in the original game are reported in Table 6). On the other hand, the column generation approaches struggle since the dimension of the pure strategy space depends exponentially on the number of information sets. The performance of our method depends crucially on having low information complexity. In fact, in games such as <sup>3</sup>K8 and <sup>3</sup>K12 where the information complexity is high, we observe poor performance even though the game tree is small. On the other hand, column generation techniques avoid this cost by considering an incrementally larger action space.

### 3.9 Conclusion

This paper proposed a novel two-player zero-sum representation called the *team-belief DAG* for the computation of mixed Nash equilibria in timeable two-player zero-sum imperfect-recall games and team max-min equilibria with correlation in adversarial team games. We proposed a conversion mechanism that can be interpreted from the point of view of a perfect-recall coordinator which manages all the player’s strategic choices while not accessing any information destined to be imperfectly recalled. The behavior of such a coordinator is defined based on beliefs and observations, novel concepts that allow an intuitive yet effective characterization. We also introduced a DAG decision problem structure for the TB-DAG to characterize more efficiently our conversion, by avoiding the pitfalls of an extensive-form characterization of the equivalent game. We theoretically analyzed the efficiency of our method through worst-case bounding of the size of the converted game, and we experimentally tested it on a set of customary benchmark games against a state-of-the-art approach from the literature. Our results are accompanied by novel complexity results that further characterize the hardness of computing equilibria in imperfect-recall games. In particular, we prove that computing a max-min strategy in behavioral strategies is  $\Sigma_2^P$ -hard even when the number of players is constant and there is no chance. Similarly, we prove that computing a Nash equilibrium in mixed strategies is  $\Delta_2^P$ -hard.

Many directions departing from this work can be interesting for further development of the literature on imperfect-recall and team games. In particular, designing an algorithm able to exploit both the TB-DAG representation and the incrementality of column generation is an interesting approach to surpass the previous developments. Moreover, the TB-DAG construction may possibly be improved by preprocessing the game to reduce its information complexity, mitigating the exponential blowup due, while generalizing the notion of *triangle-free games* (Farina and Sandholm, 2020) to DAG games may extend the class of games that can be solvable in polynomial time. Another possible direction follows the more traditional two-player zero-sum literature. It aims to develop specific abstraction, dynamic pruning, and subgame-solving techniques tailored to our conversion’s resulting two-player zero-sum games. Finally, the question whether some of the results presented in the paper can be extended to the non-timeable or absent-minded imperfect-recall case is open.

## 4 Hidden-Role Games: Equilibrium Concepts and Computation

### 4.1 Introduction

Consider a multiagent system with communication where the majority of agents share incentives, but there are also hidden defectors who seek to disrupt their progress.

This paper adopts the lens of game theory to characterize and solve a class of games called *hidden-role games*<sup>25</sup>. Hidden-role games model multi-agent systems in which a team of “good” agents work together to achieve some desired goal, but a subset of adversaries hidden among the agents seeks to sabotage the team.

---

<sup>25</sup>These games are often commonly called *social deduction games*.

Customarily (and in our paper), the “good” agents make up a majority of the players, but they will not know who the adversaries are. On the other hand, the adversaries know each other.

Hidden-role games offer a framework for developing optimal strategies in systems and applications that face deception. They have a strong emphasis on communication: players need to communicate in order to establish trust, coordinate actions, exchange information, and distinguish teammates from adversaries.

Hidden-role games can be used to model a wide range of recreational and real-world applications. Notable recreational examples include the popular tabletop games *Mafia* (also known as *Werewolf*) and *The Resistance*, of which *Avalon* is the best-known variant. As an example, consider the game *Mafia*. The players are split in an uninformed majority called *villagers* and an informed minority called *mafiosi*. The game proceeds in two alternating phases, *night* and *day*. In the night phase, the mafiosi privately communicate and eliminate one of the villagers. In the day phase, players vote to eliminate a suspect through majority voting. The game ends when one of the teams is completely eliminated.

We now provide several non-recreational examples of hidden-role games. In many cybersecurity applications (Garcia Morchon et al., 2007; Garcia-Morchon et al., 2013; Tripathi et al., 2022), an adversary compromises and controls some nodes of a distributed system whose functioning depends on cooperation and information sharing among the nodes. The system does not know which nodes have been compromised, and yet it must operate in the presence of the compromised nodes.

Another instance of problems that can be modeled as hidden-role games arises in *AI alignment*, *i.e.*, the study of techniques to steer AI systems towards humans’ intended goals, preferences, or ethical principles (Ziegler et al., 2019; Ji et al., 2023; Hubinger et al., 2024). In this setting, there is risk that a misaligned AI agent may attempt to deceive a human user into trusting its suggestions (Park et al., 2023; Scheurer et al., 2023). AI debate (Irving et al., 2018) aims at steering AI agents by using an adversarial training procedure in which a judge has to decide which is the more trustful between two hidden agents, one of which is a deceptor trained to fool the judge. Miller et al. (2021) proposes an experimental setting consisting of a chess game in which one side is controlled by a player and two advisors, which falls directly under our framework. The advisors pick action suggestions for the player to choose from, but one of the two advisors has the objective of making the team lose.

Hidden-role games also include general scenarios where agents receive inputs from other agents which may be compromised. For example, in *federated learning* (a popular category of distributed learning methods), a central server aggregates machine learning models trained by multiple distributed local agents. If some of these agents are compromised, they may send doctored input with the goal of disrupting the training process (Mothukuri et al., 2021).

Our paper aims to characterize optimal behavior in these settings, and analyze its computability.

**Related work.** To the best of our knowledge, there have been no previous works on general hidden-role games. On the other hand, there has been a limited amount of prior work on solving specific hidden-role games. Braverman et al. (2006) propose an optimal strategy for *Mafia*, and analyze the win probability when varying the number of players with different roles. Similarly, Christiano (2018) proposes a theoretical analysis for *Avalon*, investigating the possibility of *whispering*, *i.e.* any two players being able to communicate without being discovered. Both of those papers describe game-specific strategies that can be adopted by players to guarantee a specific utility to the teams. In contrast, we provide, to our knowledge, the first rigorous definition of a reasonable solution concept for hidden-role games, an algorithm to find such equilibria, and an experimental evaluation with a wide range of parameterized instances.

Deep reinforcement learning techniques have also been applied to various hidden-role games (Aitchison et al., 2021; Koppurapu et al., 2022; Serrino et al., 2019), but with no theoretical guarantees and usually with no communication allowed between players. A more recent stream of works focused on investigating the deceptive capabilities of large language models (LLMs) by having them play a hidden-role game (Xu et al., 2023; O’Gara, 2023). The agents, being LLM-based, communicate using plain human language. However, as before, these are not grounded in any theoretical framework, and indeed we will illustrate that optimal

strategies in hidden-role games are likely to involve communication that does not bear resemblance to natural language, such as the execution of cryptographic protocols.

### 4.1.1 Main Modeling Contributions

We first here give an informal, high-level description of our game model. We also introduce our main solution concept of interest, called *hidden-role equilibrium*, and discuss the challenges it addresses. We will define these concepts in more formality beginning in Section 4.3.

We define a (finite) *hidden-role game* as an  $n$ -player finite extensive-form game  $\Gamma$  in which the players are partitioned at the start of the game into two teams. Members of the same team share the same utility function, and the game is zero-sum, *i.e.* any gain for one team means a loss for the other. We thus identify the teams as  $\blacktriangle$  and  $\blacktriangledown$ , since teams share the same utility function, but have opposite objectives. At the start of the game, players are partitioned at random into two teams. A crucial assumption is that one of the two teams is *informed*, *i.e.* all the members of that particular team know the team assignment of all the players, while this is not true for all players belonging to the other team. Without loss of generality, we use  $\blacktriangle$  to refer to the uninformed team, and  $\blacktriangledown$  to refer to the informed one.<sup>26</sup>

To allow our model to cover *communication* among players, we formally define the *communication extensions* of a game  $\Gamma$ . The communication extensions are games like  $\Gamma$  except that actions allowing messages to be sent between players are explicitly encoded in the game. In the *public communication extension*, players are able to publicly broadcast messages. In the *private communication extension*, in addition to the public broadcast channel, the players have pairwise private communication channels.<sup>27</sup> In all cases, communication channels are *synchronous* and *authenticated*: messages sent on one timestep are received at the next timestep, and are tagged with their sender. Communication presents the main challenge of hidden role games:  $\blacktriangle$ -players wish to share information with teammates, but *not* with  $\blacktriangledown$ -players.

In defining communication extensions, we must bound the *length* of the communication, that is, how many rounds of communication occur in between every move of the game, and how many distinct messages can be sent on each round. To do this, we fix a finite message space<sup>28</sup> of size  $M$  and length of communication  $R$ , and in our definition of equilibrium we will take a supremum over  $M$  and  $R$ . This will allow us to consider arbitrarily complex message spaces (*i.e.*,  $M$  and  $R$  arbitrarily large) while still only analyzing finite games: for any *fixed*  $M$  and  $R$ , the resulting game is a finite hidden-role game. We will show that our positive results (upper bounds) only require  $\log M = R = \text{polylog}(|H|, 1/\epsilon)$ , where  $|H|$  is the number of nodes (histories) in the game tree and  $\epsilon$  is the desired precision of equilibrium.

We characterize optimal behavior in the hidden-role setting by converting hidden-role games into team games in a way that preserves the strategic aspect of hidden-roles. This team game is called *split-personality form* of a given hidden-role game. Given a (possibly communication-extended) hidden-role game  $\Gamma$ , we define and analyze two possible variants:

- the *uncoordinated split-personality form*  $\text{USPLIT}(\Gamma)$  is a team games with  $2n$  players, derived by splitting each player  $i$  in the original game in two distinct players,  $i^+$  and  $i^-$ , that pick actions for  $i$  in  $\Gamma$  if the player is assigned to team  $\blacktriangle$  or  $\blacktriangledown$  respectively.
- the *coordinated split-personality form*  $\text{CSPLIT}(\Gamma)$  is the  $(n+1)$ -player team game in which the additional player, who we refer to simply as the *adversary* or  $\blacktriangledown$ -*player*, takes control of the actions of all players who have been assigned to the  $\blacktriangledown$ -team. On the contrary, the players from 1 to  $n$  control the players as usual only if they belong to the team  $\blacktriangle$ .

The coordinated split-personality variant encodes an extra assumption on  $\blacktriangledown$ 's capabilities, namely, that the  $\blacktriangledown$ -team is controlled by a single player and is therefore perfectly coordinated. Trivially, when only one player

<sup>26</sup>For example, in *Mafia*, the villagers are  $\blacktriangle$  while mafiosi are  $\blacktriangledown$ .

<sup>27</sup>If players are assumed to be computationally bounded, pairwise private channels can be created from the public broadcast channels through public-key cryptography. However, throughout this paper, for the sake of conceptual cleanliness, we will not assume that players are computationally bounded, and therefore we will distinguish the public-communication case from the private-communication case.

<sup>28</sup>Note that, if the message space is of size  $M$ , a message can be sent in  $O(\log M)$  bits.

is on team  $\blacktriangledown$ , the uncoordinated and coordinated split-personality forms coincide.

In either case, the resulting game is a team game in which each player has a fixed team assignment. We remark that the split-personality form maintains the strategic aspects of hidden roles, since  $i^+$  and  $i^-$  share identity when interacting with the environment. For example, players may observe that  $i$  has done an action, but do not know if the controller was  $i^+$  or  $i^-$ . Similarly, messages sent by  $i^+$  and  $i^-$  are signed by player  $i$ , since the communication extension is applied on  $\Gamma$  *before* splitting personalities.

Picking which split-personality variant to use is a modeling assumption that depends on the game instance that one wants to address. For example, in many recreational tabletop games, USPLIT is the more reasonable choice because  $\blacktriangledown$ -players are truly distinct; however, in a network security game where a single adversary controls the corrupted nodes, CSPLIT is the more reasonable choice. The choice of which variant also affects the complexity of equilibrium computation: as we will detail in later, CSPLIT yields a more tractable solution concept. In certain special cases, however, CSPLIT and USPLIT will coincide. For example, we will later show that this is the case in *Avalon*, which is key to allowing our algorithms to work in that game.

With these pieces in place, we define the *hidden-role equilibria* (HRE) of a hidden-role game  $\Gamma$  as the *team max-min equilibria* (TMEs) of the split-personality form of  $\Gamma$ . That is, the hidden-role equilibria are the optimal joint strategies for team  $\blacktriangle$  in the split-personality game, where optimality is judged by the expected value against a jointly-best-responding  $\blacktriangledown$ -team. The *value* of a hidden-role game is the expected value for  $\blacktriangle$  in any hidden-role equilibrium. If communication (private or public) is allowed, we define hidden-role equilibria and values by taking the supremum over  $M$  and  $R$  of the expected value at the equilibrium, that is, the  $\blacktriangle$ -team is allowed to set the parameters of the communication.

Our new solution concept encodes, by design, a pessimistic assumption for the  $\blacktriangle$ -team.  $\blacktriangle$  picks  $M$ ,  $R$  and its strategy considering a worst-case  $\blacktriangledown$  adversary that knows this strategy and best-responds to it. Throughout our proofs, we will heavily make use of this fact. In particular, we will often consider  $\blacktriangledown$ -players that “pretend to be  $\blacktriangle$ -players” under certain circumstances, which is only possible if  $\blacktriangledown$ -players know  $\blacktriangle$ -players’ strategies. It is *not* allowed for  $\blacktriangle$ -players to know  $\blacktriangledown$ -players’ strategies in the same fashion. This is in stark contrast to usual zero-sum game analysis, where various versions of the *minimax theorem* promise that the game is unchanged no matter which side commits first to a strategy. Indeed, we discuss in Section 4.6.2 the fact that, for hidden-role games, the asymmetry is in some sense necessary: a minimax theorem *cannot* hold for nontrivial hidden-role games. We argue, however, that this asymmetry is natural and *inherent* in the hidden-role setting. If we assumed the contrary and inverted the order of the teams so that  $\blacktriangledown$  commits first to its strategy,  $\blacktriangle$  could discover the roles immediately by agreeing to message a passphrase unknown to  $\blacktriangledown$  in the first round, thus spoiling the whole purpose of hidden-role games. This argument will be made formal in Section 4.6.2.

**Existing solution concepts failures.** We defined our equilibrium notion as a team max-min equilibrium (TME) of the split-personality form of a communication-extended hidden role game. Here, we will argue why some other notions would be less reasonable.

- *Nash Equilibrium.* A *Nash equilibrium* (Nash, 1950) is a strategy profile for all players from which no player can improve its own utility by deviating. This notion is unsuitable for our purposes because it fails to capture *team coordination*. In particular, in pure coordination games (in which all players have the same utility function), which are a special case of hidden-role games (with no hidden roles and no adversary team at all), a Nash equilibrium would only be locally optimal in the sense that no player can improve the team’s value alone. In contrast, our notion will lead to the optimal team strategy in such games.
- *Team-correlated equilibrium.* The *team max-min equilibrium with correlation* (von Stengel and Koller, 1997; Celli and Gatti, 2018) (TMECor), is a common solution concept used in team games. It arises from allowing each team the ability to communicate in private (in particular, to generate correlated randomness) *before* the game begins. For team games, TMECor is arguably a more natural notion than TME, as the corresponding optimization problem is a bilinear saddle-point problem, and therefore in particular the minimax theorem applies, avoiding the issue of which team ought to commit first. However, for hidden-role games, TMECor is undesirable, because it does not make sense for a team

to be able to correlate with teammates that have not even been assigned yet. The *team max-min equilibrium with communication* (TMECom) (Celli and Gatti, 2018) makes an even stronger assumption about communication among team members, and therefore suffers the same problem.

### 4.1.2 Main Computational Contributions

We now introduce computational results, both positive and negative, for computing the hidden-role value and hidden-role equilibria of a given game.

**Polynomial-time algorithm.** Our main positive result is summarized in the following informal theorem statement.

**Theorem 4.1** (Main result, informal; formal result in Theorem 4.12). *If the number of players is constant, private communication is available, the  $\blacktriangledown$ -team is a strict minority (i.e., strictly less than half of the players are on the  $\blacktriangledown$ -team), and the adversary is coordinated, there is a polynomial-time algorithm for exactly computing the hidden-role value.*

This result should be surprising, for multiple reasons. First, team games are generally hard to solve (von Stengel and Koller, 1997; Zhang et al., 2023b), so any positive result for computing equilibria in team games is fairly surprising. Further, it is *a priori* not obvious that the value of any hidden-role game with private communication and coordinated adversary is even a *rational* number<sup>29</sup>, much less computable in polynomial time: for example, there exist adversarial team games with no communication whose TME values are irrational (von Stengel and Koller, 1997).

There are two key ingredients to the proof of Theorem 4.1. The first is a special type of game which we call a *mediated* game. In a mediated game, there is a player, the *mediator*, who is always on team  $\blacktriangle$ .  $\blacktriangle$ -players can therefore communicate with it and treat it as a trusted party. We show that, when a mediator is present (and all the other assumptions of Theorem 4.1 also hold), the hidden-role value is computable in polynomial time. To do this, we state and prove a form of *revelation principle*. Informally, our revelation principle states that, without loss of generality, it suffices to consider  $\blacktriangle$ -team strategies in which, at every timestep of the game,

1. all  $\blacktriangle$ -players send their honest information to the mediator,
2. the mediator sends action recommendations to all players (regardless of their team allegiance; remember that the mediator may not know the team assignment), and
3. all  $\blacktriangle$ -players play their recommended actions.

$\blacktriangledown$ -team players are, of course, free to pretend to be  $\blacktriangle$ -team players and thus send false information to the mediator; the mediator must deal with this possibility. However,  $\blacktriangledown$ -team players cannot just send *any* message; they must send messages that *are consistent with some  $\blacktriangle$ -player*, lest they be immediately revealed as  $\blacktriangledown$ -team. These observations are sufficient to construct a *two-player zero-sum game*  $\Gamma_0$ , where the mediator is the  $\blacktriangle$ -player and the coordinated adversary is the  $\blacktriangledown$ -player. The value of  $\Gamma_0$  is the value of the original hidden-role game, and the size of  $\Gamma_0$  is at most polynomially larger than the size of the original game. Since two-player zero-sum extensive-form games can be solved in polynomial time (Koller et al., 1994; von Stengel, 1996), it follows that mediated hidden-role games can also be solved in polynomial time.

The second ingredient is to invoke results from the literature on *secure multi-party computation* to *simulate* a mediator in the case that one is not already present. A well-known result from that literature states that so long as strictly more than half of the players are honest, essentially any interactive protocol—such as the ones used by our mediator to interact with other players—can be simulated efficiently such that the adversary can cause failure of the protocol or leakage of information (Beaver, 1990; Rabin and Ben-Or, 1989).<sup>30</sup> Chaining

<sup>29</sup>assuming all game values and chance probabilities are also rational numbers

<sup>30</sup>In this part of the argument, the details about the communication channels become important: in particular, the MPC results that we use for our main theorem statement assume that the network is synchronous (*i.e.*, messages sent in round  $r$  arrive in round  $r + 1$ ), and that there are pairwise private channels and a public broadcast channel that are all authenticated (*i.e.*, message receivers know who sent the message). This is enough to implement MPC so long as  $k < n/2$ , where  $k$  is the



such a protocol with the argument in the previous paragraph concludes the proof of the main theorem.

**Related works on MPC and communication equilibria.** The *communication equilibrium* (Forges, 1986; Myerson, 1986) is a notion of equilibrium with a mediator, in which the mediator has two-way communication with all players, and players need to be incentivized to report information honestly and follow recommendations. Communication equilibria include all Nash equilibria, and therefore are unfit for general hidden-role games for the same reason as Nash equilibria, as discussed in the previous subsection.

However, when team  $\blacktriangledown$  has only one player and private communication is allowed, the hidden-role equilibria coincide with the  $\blacktriangle$ -team-optimal communication equilibria in the original game  $\Gamma$ . Our main result covers this case, but an alternative way of computing a hidden-role equilibrium in this special case is to apply the optimal communication equilibrium algorithms of Zhang and Sandholm (2022a) or Zhang et al. (2023a). However, those algorithms either involve solving linear programs, solving many zero-sum games, or solving zero-sum games with large reward ranges, which will be less efficient than directly solving a single zero-sum game  $\Gamma_0$ .

We are not the first to observe that multi-party computation can be used to implement a mediator for use in game theory. In various settings and for various solution concepts, it is known to be possible to implement a mediator using only cheap-talk communication among players [e.g., Urbano and Vila, 2002; Liu, 2017; Abraham et al., 2006; Izmalkov et al., 2005]. For additional reading on the connections between game theory and cryptography, we refer the reader to the survey of Katz (2008), and papers citing and cited by this survey.

**Lower bounds.** We also show *lower bounds* on the complexity of computing the hidden-role value, even for a constant number of players, when any of the assumptions in Theorem 4.1 is broken.

**Theorem 4.2** (Lower bounds, informal; formal statement in Theorems 4.14 and 4.15). *If private communication is disallowed, the hidden-role value problem is NP-hard. If the  $\blacktriangledown$ -team is uncoordinated, the problem is coNP-hard. If both, the problem is  $\Sigma_2^P$ -hard. All hardness reductions hold even when the  $\blacktriangledown$ -team is a minority and the number of players is an absolute constant.*

**Price of hidden roles.** Finally, we define and compute the *price of hidden roles*. It is defined (analogously to the price of anarchy and price of stability, which are common quantities of study in game theory) as the ratio between the value of a hidden-role game, and the value of the same game with team assignments made public. We show the following:

**Theorem 4.3** (Price of hidden roles; formal statement in Theorem 4.19). *Let  $D$  be a distribution of team assignments. For the class of games where teams are drawn according to distribution  $D$ , the price of hidden roles is equal to  $1/p$ , where  $p$  is the probability of the most-likely team in  $D$ .*

Intuitively, in the worst case, the  $\blacktriangle$ -team can be forced to guess at the beginning of the game all the members of the  $\blacktriangle$ -team, and win if and only if its guess is correct. In particular, for the class of  $n$ -player games with  $k$  adversaries, the price of hidden roles is exactly  $\binom{n}{k}$ .

---

number of adversaries and  $n$  is the number of players. Our results, however, do not depend on the specific assumptions about the communication channel, so long as said assumptions enable secure MPC with guaranteed outcome delivery. For a recent survey of MPC, see Lindell (2020). For example, if  $k < n/3$  then MPC does not require a public broadcast channel, so neither do our results. For cleanliness, and to avoid introducing extra formalism, we will stick to one model of communication.

### 4.1.3 Experiments: *Avalon*

We ran experiments on the popular tabletop game *The Resistance: Avalon* (or simply *Avalon*, for short). As discussed earlier, despite the adversary team in *Avalon* not being coordinated in the sense used in the rest of the paper, we show that, at least for the 5- and 6-player variants, the adversary would not benefit from being coordinated; therefore, our polynomial-time algorithms can be used to solve the game. This observation ensures that our main result applies. Game-specific simplifications allow us to reduce the game tree from roughly  $10^{56}$  nodes (Serrino et al., 2019) to  $10^8$  or even fewer, enabling us to compute exact equilibria. Our experimental evaluation demonstrates the practical efficacy of our techniques on real game instances. Our results are discussed in Section 4.7, and further detail on the game-specific reductions used, as well as a complete hand-analysis of a small *Avalon* variant, can be found in the appendix of the full paper (Carminati et al., 2024b).

### 4.1.4 Examples

In this section we present three examples that will hopefully help the reader in understanding our notion of equilibrium and justify some choices we have made in our definition.

**A hidden-role matching pennies game.** Consider a  $n$ -player version of matching pennies (with  $n > 2$ ), which we denote as  $\text{MP}(n)$ . One player is chosen at random to be the adversary (team  $\blacktriangledown$ ). All  $n$  players then simultaneously choose bits  $b_i \in \{0, 1\}$ . Team  $\blacktriangle$  wins (gets utility 1) if and only if all  $n$  bits match; else, team  $\blacktriangle$  loses (gets utility 0).

With no communication, the value of this game is  $1/2^{n-1}$ : it is an equilibrium for everyone to play uniformly at random. Public communication does not help, because, conditioned on the public transcript, bits chosen by players must be mutually independent. Thus, the adversary can do the following: pretend to be on team  $\blacktriangle$ , wait for all communication to finish, and then play 0 if the string of all ones is more conditionally likely than the string of all 1s, and vice-versa.

With private communication, however, the value becomes  $1/(n+1)$ . Intuitively, the  $\blacktriangle$ -team should attempt to guess who the  $\blacktriangledown$ -player is, and then privately discuss among the remaining  $n-1$  players what bit to play. We defer formal proofs of the above game values to Section 4.5, because they rely on results in Section 4.4.1.

**Simultaneous actions.** In typical formulations of extensive-form imperfect-information games, it is without loss of generality to assume that games are *turn-based*, *i.e.*, only one player acts at any given time. To simulate simultaneous actions with sequential ones, one can simply forbid players from observing each others' actions. However, when communication is allowed arbitrarily throughout the game, the distinction between simultaneous and sequential actions suddenly becomes relevant, because *players can communicate when one—but not all—the players have decided on an action*.

To illustrate this, consider the game  $\text{MP}(n)$  defined in the previous section, with public communication, except that the players act in sequence in order of index  $(1, 2, \dots, n)$ . We claim that the value of this game is not  $1/2^{n-1}$ , but at least  $1/2n$ . To see this, consider the following strategy for team  $\blacktriangle$ . The  $\blacktriangle$  players wait for P1 to (privately) pick an action. Then, P2 publicly declares a bit  $b \in \{0, 1\}$ , and all remaining players play  $b$  if they are on team  $\blacktriangle$ . If P1 was the  $\blacktriangledown$  player, this strategy wins with probability at least  $1/2$ , so the expected value is at least  $1/2n$ . This example illustrates the importance of allowing simultaneous actions in our game formulations.

*Correlated randomness matters.* We use our third and final example to discuss a nontrivial consequence of the definition of hidden-role equilibrium that may appear strange at first: it is possible for seemingly-useless information to affect strategic decisions and the game value.

To illustrate, consider the following simple game  $\Gamma$ : there are three players, and three role cards. Two of the three cards are marked  $\blacktriangle$ , and the third is marked  $\blacktriangledown$ . The cards are dealt privately and randomly to the players. Then, after some communication, all three players simultaneously cast votes to elect a winner. If no player gains a majority of votes,  $\blacktriangledown$  wins. Otherwise, the elected winner’s team wins. Clearly,  $\blacktriangle$  can win no more than  $2/3$  of the time in this game:  $\blacktriangledown$  can simply pretend to be on team  $\blacktriangle$ , and in that case  $\blacktriangle$  cannot gain information, and the best they can do is elect a random winner.

Now consider the following seemingly-meaningless modification to the game. We will modify the two  $\blacktriangle$  cards so that they are distinguishable. For example, one card has  $\blacktriangle$  written on it, and the other has  $\blacktriangle'$ . We argue that this, perhaps surprisingly, affects the value of the game. In fact, the  $\blacktriangle$  team can now win deterministically, even with only public communication. Indeed, consider following strategy. The two players on  $\blacktriangle$  publicly declare what is written on their cards (*i.e.*,  $\blacktriangle$  or  $\blacktriangle'$ ). The player elected now depends on what the third player did. If one player does not declare  $\blacktriangle$  or  $\blacktriangle'$ , elect either of the other two players. If two players declared  $\blacktriangle$ , elect the player who declared  $\blacktriangle'$ . If two players declare  $\blacktriangle'$ , elect the player who declared  $\blacktriangle$ . This strategy guarantees a win: no matter what the  $\blacktriangledown$ -player does, any player who makes a unique declaration is guaranteed to be on the  $\blacktriangle$ -team.

What happens in the above example is that making the cards distinguishable introduces a piece of *correlated randomness* that  $\blacktriangle$  can use: the two  $\blacktriangle$  players receive cards whose labels are (perfectly negatively) correlated with each other. Since our definition otherwise prohibits the use of such correlated randomness (because players cannot communicate only with players on a specific team), introducing some into the game can have unintuitive effects. In Section 4.6.2, we expand on the effects of allowing correlated randomness: in particular, we argue that allowing correlated randomness essentially ruins the point of hidden-role games by allowing the  $\blacktriangle$  team to learn the entire team assignment.

## 4.2 Preliminaries

Our notation in this part differs slightly from other parts of this thesis. For reasons alluded to in the introduction, we explicitly allow simultaneous moves in our formulation. More specifically, at each history  $h \in \mathcal{H} \setminus \mathcal{Z}$ , every player (including chance) selects an action  $a \in \mathcal{A}_i(h)$ , and the edges leaving  $h$  are identified with *joint actions*  $a \in \times_{i \in [n] \cup C} \mathcal{A}_i(h)$ . Thus, each player’s infoset partition  $\mathcal{I}$  is a partition of  $\mathcal{H} \setminus \mathcal{Z}$ .

An extensive-form game is an *adversarial team game* (ATG) if there is a team assignment  $t \in \{\blacktriangle, \blacktriangledown\}^n$  and a team utility function  $u : \mathcal{Z} \rightarrow \mathbb{R}$  such that  $u_i(z) = u(z)$  if  $t_i = \blacktriangle$ , and  $u_i(z) = -u(z)$  if  $t_i = \blacktriangledown$ . That is, each player is assigned to a team, all members of the team get the same utility, and the two teams are playing an adversarial zero-sum game<sup>31</sup>. In this setting, we will write  $\mathbf{x}_i \in \mathcal{X}_i$  and  $\mathbf{y}_j \in \mathcal{Y}_j$  for a generic strategy of a player on team  $\blacktriangle$  and  $\blacktriangledown$  respectively. ATGs are fairly well studied. In particular, Team Maxmin Equilibria (TMEs) (von Stengel and Koller, 1997; Celli and Gatti, 2018) and their variants are the common notion of equilibrium employed. The *value* of a given strategy profile  $\mathbf{x}$  for team  $\blacktriangle$  is the value that  $\mathbf{x}$  achieves against a best-responding opponent team. The *TME value* is the value of the best strategy profile of team  $\blacktriangle$ . That is, the TME value is defined as

$$\text{TMEVal}(\Gamma) := \max_{\mathbf{x} \in \times_i \Delta(\mathcal{X}_i)} \min_{\mathbf{y} \in \times_j \Delta(\mathcal{Y}_j)} u(\mathbf{x}, \mathbf{y}), \quad (4)$$

and the *TMEs* are the strategy profiles  $\mathbf{x}$  that achieve the maximum value. Notice that the TME problem is nonconvex, since the objective function  $u$  is nonlinear as a function of  $\mathbf{x}$  and  $\mathbf{y}$ . As such, the minimax theorem does not apply, and swapping the teams may not preserve the solution. Computing an (approximate) TME is  $\Sigma_2^P$ -complete in extensive-form games (Zhang et al., 2023b).

<sup>31</sup>This is a slight abuse of language: if the  $\blacktriangle$  and  $\blacktriangledown$  teams have different sizes, the sum of all players’ utilities is not zero. However, such a game can be made zero-sum by properly scaling each player’s utility. The fact that such a rescaling operation does not affect optimal strategies is a basic result for von Neumann–Morgenstern utilities (Maschler et al., 2020, Chapter 2.4). We will therefore generally ignore this detail.

### 4.3 Equilibrium Concepts for Hidden-Role Games

While the notion of TME is well-suited for ATGs, it is not immediately clear how to generalize it to the setting of hidden-role games. We do so by formally defining the concepts of *hidden-role game*, *communication* and *split-personality form* first introduced in Section 4.1.1.

**Definition 4.4.** An extensive-form game is a *zero-sum hidden-role team game*, or *hidden-role game* for short, if it satisfies the following additional properties:

1. At the root node, only chance has a nontrivial action set. Chance chooses a string  $t \sim \mathcal{D} \in \Delta(\{\blacktriangle, \blacktriangledown\}^n)$ , where  $t_i$  denotes the team to which player  $i$  has been assigned. Each player  $i$  privately observes (at least<sup>32</sup>) their team assignment  $t_i$ . In addition,  $\blacktriangledown$ -players privately observe the entire team assignment  $t$ .
2. The utility of a player  $i$  is defined completely by its team: there is a  $u : \mathcal{Z} \rightarrow \mathbb{R}$  for which  $u_i(z) = u(z)$  if player  $i$  is on team  $\blacktriangle$  at node  $z$ , and  $-u(z)$  otherwise.<sup>33</sup>

In some games, players observe additional information beyond just their team assignments. For example, in *Avalon*, one  $\blacktriangle$ -player is designated *Merlin*, and Merlin has additional information compared to other  $\blacktriangle$ -players. In such cases, we will distinguish between the *team assignment* and *role* of a player: the team assignment is just the team that the player is on ( $\blacktriangle$  or  $\blacktriangledown$ ), while the role encodes the extra private information of the player as well, which may affect what actions that player is allowed to legally take. For example, the team assignment of the player with role *Merlin* is  $\blacktriangle$ . We remark that additional imperfect information of the game may be observed after the root node.<sup>34</sup>

Throughout this paper, we will use  $k$  to denote the largest number of players on the  $\blacktriangledown$ -team, that is,  $k = \max_{t \in \text{supp}(\mathcal{D})} |\{i : t_i = \blacktriangledown\}|$ .

#### 4.3.1 Models of Communication

The bulk of this paper concerns notions of equilibrium that allow communication between the players. We distinguish in this paper between *public* and *private communication*:

1. *Public communication*: There is an open broadcast channel on which all players can send messages.
2. *Private communication*: In addition to the open broadcast channel, each pair of players has access to a private communication channel. The private communication channel reveals to all players when messages are sent, but only reveals the message contents to the intended recipients.

Assuming that public-key cryptography is possible (*e.g.*, assuming the discrete logarithm problem is hard) and players are polynomially computationally bounded, public communication and private communication are equivalent, because players can set up pairwise private channels via public-key exchange. However, in this paper, we assume that agents are computationally unbounded and thus treat the public and private communication cases as different. Our motivation for making this distinction is twofold. First, it is conceptually cleaner to explicitly model private communication, because then our equilibrium notion definitions do not need to reference computational complexity. Second, perhaps counterintuitively, equilibria with public communication only may be *more* realistic to execute in practice in human play, precisely *because* public-key cryptography breaks. That is, the computationally unbounded adversary renders more “complex” strategies of the  $\blacktriangle$ -team (involving key exchanges) useless, thus perhaps resulting in a *simpler* strategy. We emphasize that, in all of our positive results in the paper, the  $\blacktriangle$ -team’s strategy *is* efficiently computable.

To formalize these notions of communication, we now introduce the *communication extension*.

<sup>32</sup>It is allowable for  $\blacktriangle$ -players to also have more observability of the team assignment, *e.g.*, certain  $\blacktriangle$ -players may know who some  $\blacktriangledown$ -players are.

<sup>33</sup>While at a first look this condition is similar to the one in ATGs, we remark that in this case the number of players in a team depends on the roles assigned at the start. The same considerations as Footnote 31 on the zero-sum rescaling of the utilities hold.

<sup>34</sup>This is an important difference with respect to Bayesian games (Harsanyi, 1967–68), which assume all imperfect information to be the initial *types* of the players. Conversely, we have an imperfect information structure that evolves throughout the game, while only the teams are assigned and observed at the start.

**Definition 4.5.** The *public* and *private*  $(M, R)$ -communication extensions corresponding to a hidden-role game  $\Gamma$  are defined as follows. Informally, between every step of the original game  $\Gamma$ , there will be  $R$  rounds of communication; in each round, players can send a public broadcast message and private messages to each player. The communication extension starts in state  $h = \emptyset \in \mathcal{H}_\Gamma$ . At each game step of  $\Gamma$ :

1. Each player  $i \in [n]$  observes its info set  $I_i \ni h$ .
2. For each of  $R$  successive communication rounds:
  - (a) Each player  $i$  simultaneously chooses a message  $m_i \in [M]$  to broadcast publicly.
  - (b) If private communication is allowed, each player  $i$  also chooses messages  $m_{i \rightarrow j} \in [M] \cup \{\perp\}$  to send to each player  $j \neq i$ .  $\perp$  denotes that the player does not send a private message at that time.
  - (c) Each player  $j$  observes the messages  $m_{i \rightarrow j}$  that were sent to it, as well as all messages  $m_i$  that were sent publicly. That is, by notion of communication, the players observe:
    - *Public*: player  $j$  observes the ordered tuple  $(m_1, \dots, m_n)$ .
    - *Private*: player  $j$  also observes the ordered tuple  $(m_{1 \rightarrow j}, \dots, m_{n \rightarrow j})$ , and the set  $\{(i, k) : m_{i \rightarrow k} \neq \perp\}$ . That is, players observe messages sent to them, and players see when other players send private messages to each other (but not the contents of those messages)
3. Each player, including chance, simultaneously plays an action  $a_i \in \mathcal{A}_i(h)$ . (Chance plays according to its fixed strategy.) The game state  $h$  advances accordingly.

We denote the  $(M, R)$ -extensions as  $\text{COMM}_{\text{priv}}^{M,R}(\Gamma)$ , and  $\text{COMM}_{\text{pub}}^{M,R}(\Gamma)$ . To unify notation, we also define  $\text{COMM}_{\text{none}}^{M,R}(\Gamma) = \Gamma$ . When the type of communication allowed and number of rounds are not relevant, we use  $\text{COMM}(\Gamma)$  to refer to a generic extension.

### 4.3.2 Split Personalities

We introduce two different *split-personality* forms  $\text{USPLIT}(\Gamma)$  and  $\text{CSPLIT}(\Gamma)$  of a hidden-role game  $\Gamma$ . The split-personality forms are adversarial team games which preserve the characteristics of  $\Gamma$ .

**Definition 4.6.** The *uncoordinated split-personality form*<sup>35</sup> of an  $n$ -player hidden-role game  $\Gamma$  is the  $2n$ -player adversarial team game  $\text{USPLIT}(\Gamma)$  in which each player  $i$  is split into two players,  $i^+$  and  $i^-$ , which control player  $i$ 's actions when  $i$  is on team  $\blacktriangle$  and team  $\blacktriangledown$  respectively.

Unlike the original hidden-role game  $\Gamma$ , the split-personality game is an adversarial team game without hidden roles: players  $i^+$  are on the  $\blacktriangle$  team, and  $i^-$  are on the  $\blacktriangledown$ -team. Therefore, we are able to apply notions of equilibrium for ATGs to  $\text{USPLIT}(\Gamma)$ . We also define the *coordinated split-personality form*:

**Definition 4.7.** The *coordinated split-personality form* of an  $n$ -player hidden-role game  $\Gamma$  is the  $(n + 1)$ -player adversarial team game  $\text{CSPLIT}(\Gamma)$  formed by starting with  $\text{USPLIT}(\Gamma)$  and merging all  $\blacktriangledown$ -players into a single adversary player, who observes all their observations and chooses all their actions.

Assuming  $\blacktriangledown$  to be *coordinated* is a worst-case assumption for team  $\blacktriangle$ , which however can be justified. In many common hidden-role games, such as the *Mafia* or *Werewolf* family of games and most variants of *Avalon*, such an assumption is not problematic, because the  $\blacktriangledown$ -team has essentially perfect information already. In the appendix of the full paper (Carminati et al., 2024b), we justify why this assumption is safe also in some more complex *Avalon* instances considered. The coordinated split-personality form will be substantially easier to analyze, and in light of the above equivalence for games like *Avalon*, we believe that it is important to study it.

When team  $\blacktriangledown$  in  $\Gamma$  is already coordinated, that is, if every  $\blacktriangledown$ -team member has the same observation at every timestep, the coordinated and uncoordinated split-personality games will, for all our purposes, coincide: in this case, any strategy of the adversary in  $\text{CSPLIT}(\Gamma)$  can be matched by a joint strategy of the  $\blacktriangledown$ -team members in  $\text{USPLIT}(\Gamma)$ . This is true in particular if there is only one  $\blacktriangledown$ -team member. But, we insert here a

<sup>35</sup>In the language of Bayesian games, the split-personality form would almost correspond to the *agent form*.

warning: even when the base game  $\Gamma$  has a coordinated adversary team, the private communication extension  $\text{COMM}_{\text{priv}}(\Gamma)$  will not. Thus, with private-communication extensions of  $\Gamma$ , we must distinguish the coordinated and uncoordinated split-personality games even if  $\Gamma$  itself is coordinated.

### 4.3.3 Equilibrium Notions

We now define the notions of equilibrium that we will primarily study in this paper.

**Definition 4.8.** The *uncoordinated value* of a hidden-role game  $\Gamma$  with notion of communication  $c$  is defined as

$$\text{UVal}_c(\Gamma) := \sup_{M,R} \text{UVal}_c^{M,R}(\Gamma)$$

where  $\text{UVal}_c^{M,R}(\Gamma)$  is the TME value of  $\text{USPLIT}(\text{COMM}_c^{M,R}(\Gamma))$ . The *coordinated value*  $\text{CVal}_c(\Gamma)$  is defined analogously by using  $\text{CSPLIT}$ .

**Definition 4.9.** An  $\epsilon$ -*uncoordinated hidden-role equilibrium* of  $\Gamma$  with a particular notion of communication  $c \in \{\text{none}, \text{pub}, \text{priv}\}$  is a tuple  $(M, R, \mathbf{x})$  where  $\mathbf{x}$  is a  $\blacktriangle$ -strategy profile in  $\text{USPLIT}(\text{COMM}_c^{M,R}(\Gamma))$  of value at least  $\text{UVal}_c(\Gamma) - \epsilon$ . The  $\epsilon$ -*coordinated hidden-role equilibria* is defined analogously, again with  $\text{CSPLIT}$  and  $\text{CVal}$  instead of  $\text{USPLIT}$  and  $\text{UVal}$ .

As discussed in Section 4.1.1, our notion of equilibrium is inherently asymmetric due to its max-min definition. The  $\blacktriangle$ -team is the first to commit to a strategy and a communication scheme, and  $\blacktriangledown$  is allowed to know both how much communication will be used (*i.e.*,  $M$  and  $R$ ) as well as  $\blacktriangle$ 's entire strategy  $\mathbf{x}$ . As mentioned before, this asymmetry is fundamental in our setting, and we will formalize it in Section 4.6.2.

## 4.4 Computing Hidden-Role Equilibria

In this section, we show the main computational results regarding the complexity of computing an hidden-role equilibrium in different settings. We first provide positive results for the private-communication case in Section 4.4.1 while the negative computational results for the no/public-communications cases are presented in Section 4.4.2. The results are summarized in Table 8.

### 4.4.1 Computing Private-Communication Equilibria

In this section, we show that it is possible under some assumptions to compute equilibria efficiently for hidden-role games. In particular, in this section, we assume that

1. there is private communication,
2. the adversary is coordinated, and
3. the adversary is a minority ( $k < n/2$ ).

**Games with a publicly-known  $\blacktriangle$ -player.** First, we consider a special class of hidden-role games which we call *mediated*. In a mediated game, there is a player, who we call the *mediator*, who is always assigned to team  $\blacktriangle$ . The task of the mediator is to coordinate the actions and information transfer of team  $\blacktriangle$ . Our main result of this subsection is the following:

**Theorem 4.10** (Revelation Principle). *Let  $\Gamma^*$  be a mediated hidden-role game. Then, for  $R \geq 2$  and  $M \geq |H|$ , there exists a coordinated private-communication equilibrium in which the players on  $\blacktriangle$  have a TME profile in which, at every step, the following events happen in sequence:*

1. every player on team  $\blacktriangle$  sends its observation privately to the mediator,
2. the mediator sends to every player ( $\blacktriangle$  and  $\blacktriangledown$ ) a recommended action, and
3. all players on team  $\blacktriangle$  play their recommended actions.

Players on team  $\blacktriangledown$ , of course, can (and will) lie or deviate from recommendations as they wish. The above revelation principle implies the following algorithmic result:

**Theorem 4.11.** *Let  $\Gamma^*$  be a mediated hidden-role game,  $R \geq 2$ , and  $M \geq |H|$ . An (exact) coordinated private-communication hidden-role equilibrium of  $\Gamma^*$  can be computed by solving an extensive-form zero-sum game  $\Gamma_0$  with at most  $|H|^{k+1}$  nodes, where  $H$  is the history set of  $\Gamma^*$ .*

Proofs of Theorems 4.10 and 4.11 can be found in the appendix of the full paper (Carminati et al., 2024b).

We give a sketch of how the two-player zero-sum game is structured. Theorem 4.10 allows us to simplify the game by fixing the actions of all players on team  $\blacktriangle$ , leaving two strategic players, the mediator and the adversary. Any node from the original game is expanded into three levels:

1. the adversary picks messages on behalf of all  $\blacktriangledown$ -players to send to the mediator,
2. the mediator picks recommended actions to send to all players, and
3. the adversary acts on behalf of all  $\blacktriangledown$ -players.

The key to proving Theorem 4.11 is that, in the first step above, the adversary’s message space is not too large. Indeed, any message sent by the adversary must be a message that *could have plausibly been sent by a  $\blacktriangle$ -player*: otherwise the mediator could automatically infer that the sender must be the adversary. It is therefore possible to exclude all other messages from the game since they belong to dominated strategies. Carefully counting the number of such messages would complete the proof.

It is crucial in the above argument that the  $\blacktriangledown$ -team is coordinated; indeed, otherwise, it would not be valid to model the  $\blacktriangledown$ -team as a single adversary in  $\Gamma_0$ . For more elaboration on the case where the  $\blacktriangledown$ -team is not coordinated, we refer the reader to the appendix of the full paper (Carminati et al., 2024b).

In practice, zero-sum extensive-form games can be solved very efficiently in the tabular setting with linear programming (Koller et al., 1994), or algorithms from the counterfactual regret minimization (CFR) family (Brown and Sandholm, 2019a; Farina et al., 2021c; Zinkevich et al., 2007). Thus, Theorem 4.11 gives an efficient algorithm for solving hidden-role games with a mediator.

**Simulating mediators with multi-party computation.** In this section, we show that the previous result essentially generalizes (up to exponentially-small error) to games *without* a mediator, so long as the  $\blacktriangledown$  team is also a minority, that is,  $k < n/2$ . Informally, the main result of this subsection states that, when private communication is allowed, one can efficiently *simulate* the existence of a mediator using secure multi-party computation (MPC), and therefore team  $\blacktriangle$  can achieve the same value. The form of secure MPC that we use is *information-theoretically* secure; that is, it is secure even against computationally-unbounded adversaries.

**Theorem 4.12 (Main theorem).** *Let  $\Gamma$  be a hidden-role game with  $k < n/2$ . Then  $\text{CVal}_{\text{priv}}(\Gamma) = \text{CVal}_{\text{priv}}(\Gamma^*)$ , where  $\Gamma^*$  is  $\Gamma$  with a mediator added, and moreover this value can be computed in  $|H|^{O(k)}$  time by solving a zero-sum game of that size. Moreover, an  $\epsilon$ -hidden-role equilibrium with private communication and  $\log M = R = \text{polylog}(|H|, 1/\epsilon)$  can be computed and executed by the  $\blacktriangle$ -players in time  $\text{poly}(|H|^k, \log(1/\epsilon))$ .*

The proof uses MPC to simulate the mediator and then executes the equilibrium given by Theorem 4.11. The proof of Theorem 4.12, as well as requisite background on multi-party computation, are deferred to the appendix of the full paper (Carminati et al., 2024b). We emphasize that Theorems 4.11 and 4.12 are useful not only for algorithmically computing an equilibrium, but also for manual analysis of games: instead of analyzing the infinite design space of possible messaging protocols, it suffices to analyze the finite zero-sum game  $\Gamma_0$ . Our experiments on *Avalon* use both manual analysis and computational equilibrium finding algorithms to solve instances.

Adversary Team Assumptions	Communication Type		
	None	Public	Private
Coordinated, Minority	NP-complete	NP-hard	P [Thm. 4.12]
Coordinated	(von Stengel and Koller, 1997)	[Thm. 4.14]	open problem
Minority	$\Sigma_2^P$ -complete	$\Sigma_2^P$ -hard	coNP-hard
None	[Thm. 4.15] and (Zhang et al., 2023b)	[Thm. 4.15]	[Thm. 4.15]

**Table 8:** Complexity results for computing hidden-role value with a constant number of players, for various assumptions about the adversary team and notions of communication. The results shaded in green are new to our paper.

**Comparison with communication equilibria.** As mentioned in Section 4.1.2, our construction simulating a mediator bears resemblance to the construction used to define *communication equilibria* (Forges, 1986; Myerson, 1986). At a high level, a communication equilibrium of a game  $\Gamma$  is a Nash equilibrium of  $\Gamma$  augmented with a mediator that is playing according to some fixed strategy  $\mu$ . Indeed, when team  $\blacktriangledown$  has only one player, it turns out that the two notions coincide:

**Theorem 4.13.** *Let  $\Gamma$  be a hidden-role game with  $k = 1$ . Then  $\text{CVal}_{\text{priv}}(\Gamma)$  is exactly the value for  $\blacktriangle$  of the  $\blacktriangle$ -optimal communication equilibrium of  $\Gamma$ .*

However, in the more general case where  $\blacktriangledown$  can have more than one player, Theorem 4.13 does not apply: in that case, communication equilibria include all Nash equilibria in particular, and therefore fail to enforce *joint* optimality of the  $\blacktriangledown$ -team, so our concepts and methods are more suitable. The proof is deferred to the appendix of the full paper (Carminati et al., 2024b).

#### 4.4.2 Computing No/Public-Communication Equilibria

In this section, we consider games with no communication or with public-communication and a coordinated minority. Conversely to the private-communication case of Section 4.4.1, in this case the problem of computing the value of a hidden-role equilibrium is in general NP-hard.

For the remainder of this section, when discussing the problem of “computing the value of a game”, we always mean the following promise problem: given a game, a threshold  $v$ , and an allowable error  $\epsilon > 0$  (both expressed as rational numbers), decide whether the hidden-role value of  $\Gamma$  is  $\geq v$  or  $\leq v - \epsilon$ .

**Theorem 4.14.** *Even in 2-vs-1 games with public roles and  $\epsilon = 1/\text{poly}(|H|)$ , computing the TME value (and hence also the hidden-role value, since adversarial team games are a special case of hidden-role games) with public communication is NP-hard.*

Since there is only one  $\blacktriangledown$ -player in the above reduction, the result applies regardless of whether the adversary is coordinated.

**Theorem 4.15.** *Even with a constant number of players, a minority adversary team, and  $\epsilon = 1/\text{poly}(|H|)$ , computing the uncoordinated value of a hidden-role game is coNP-hard with private communication and  $\Sigma_2^P$ -hard with public communication or no communication.*

Proofs of results in this section are deferred to the appendix of the full paper (Carminati et al., 2024b). Intuitively, the proofs work by constructing gadgets that prohibit any useful communication, thus reducing to the case of no communication.



## 4.5 Worked Example

This section includes a worked example of value computation to illustrate the differences among the notions of equilibrium discussed in the paper and illustrates the utility of having a mediator for private communication. Consider a  $n$ -player version of matching pennies  $\text{MP}(n)$  as defined in Section 4.1.4.

**Proposition 4.16.** *Let  $\text{MP}(n)$  be the  $n$ -player matching pennies game.*

1. *The  $\text{TMECor}$  and  $\text{TMECom}$  values of  $\text{PUBLICTEAM}(\text{MP}(n))$  are both  $1/2$ .*
2. *Without communication or with only public communication, the value of  $\text{MP}(n)$  is  $1/2^{n-1}$ .*
3. *With private communication, the value of  $\text{MP}(n)$  is  $1/(n+1)$ .*

*Proof.* The first claim, as well as the no-communication value, is known (Basilico et al., 2017).

For the public-communication value, observe that, conditioned on the transcript, the bits chosen by the players must be mutually independent of each other. Thus, the adversary can do the following: pretend to be on team  $\blacktriangle$ , wait for all communication to finish, and then play 0 if the string of all ones is more conditionally likely than the string of all 1s, and vice-versa<sup>36</sup>.

It thus only remains to prove the third claim.

(*Lower bound*) The players simulate a mediator using multi-party computation (see Theorems 4.11 and 4.12). Consider the following strategy for the mediator. Sample a string  $b \in \{0, 1\}^n$  uniformly at random from the set of  $2n + 2$  strings that has at most one mismatched bit. Recommend to each player  $i$  that they play  $b_i$ .

Consider the perspective of the adversary. The adversary sees only a recommended bit  $b_i$ . Assume WLOG that  $b_i = 0$ . Then there are  $n + 1$  possibilities:

1.  $b$  is all zeros (1 way)
2. All other bits of  $b$  are 1 (1 way)
3. Exactly one other bit of  $b$  is 1 ( $n - 1$  ways).

The adversary wins in the third case automatically (since the team has failed to coordinate), and, regardless of what the adversary does, it can win only one of the first two cases. Thus the adversary can win at most  $n/(n+1)$  of the time, that is, this strategy achieves value  $1/(n+1)$ .

(*Upper bound*) Consider the following adversary strategy. The adversary communicates as it would do if it were on team  $\blacktriangle$ . Let  $b_i$  be the bit that the adversary would play if it were on team  $\blacktriangle$ . The adversary plays  $b_i$  with probability  $1/(n+1)$  and  $1 - b_i$  otherwise. We need only show that no pure strategy of the mediator achieves value better than  $1/(n+1)$  against this adversary. A strategy of the mediator is identified by a bitstring  $b$ . If  $b$  is all zeros or all ones, the team wins if and only if the adversary plays  $b_i$  (probability  $1/(n+1)$ ). If  $b$  has a single mismatched bit, the team wins if and only if the mismatched bit is the adversary (probability  $1/n$ ) and the adversary flips  $b_i$  (probability  $n/(n+1)$ ).  $\square$

<sup>36</sup>In general, computing the conditional probabilities could take exponential time, but when defining the notion of value here, we are assuming that players have unbounded computational resources. This argument not work for computationally-bounded adversaries. Indeed, if the adversary were computationally bounded,  $\blacktriangle$  would be able to use cryptography to build private communication channels and thus implement a mediator, allowing our main positive result Theorem 4.12 to apply.

## 4.6 Properties of Hidden-role Equilibria

In the following, we discuss interesting properties of hidden-role equilibria given the definition we provided in Section 4.1.1, and that make them fairly unique relative to other notions of equilibrium in team games.

### 4.6.1 The Price of Hidden Roles

One interesting question arising from hidden-role games is the *price* of having them. That is, how much value does  $\blacktriangle$  lose because roles are hidden? In this section, we define this quantity and derive reasonably tight bounds on it.

**Definition 4.17.** The *public-team refinement* of an  $n$ -player hidden-role game  $\Gamma$  is the adversarial team game  $\text{PUBLICTEAM}(\Gamma)$  defined by starting with the (uncoordinated) split-personality game, and adding the condition that all team assignments  $t_i$  are publicly observed by all players.

**Definition 4.18.** For a given hidden-role game  $\Gamma$  in which  $\blacktriangle$  is guaranteed a nonnegative value (*i.e.*,  $u_i(z) \geq 0$  whenever  $i$  is on team  $\blacktriangle$ ), the *price of hidden roles*  $\text{PoHR}(\Gamma)$  is the ratio between the TME value of  $\text{PUBLICTEAM}(\Gamma)$  and the hidden-role value of  $\text{USPLIT}(\Gamma)$ .

For a given class of hidden-role games  $\mathcal{G}$ , the price of hidden roles  $\text{PoHR}(\mathcal{G})$  is the supremum of the price of hidden roles across all games  $\Gamma \in \mathcal{G}$ .

**Theorem 4.19.** Let  $D \in \Delta(\{\blacktriangle, \blacktriangledown\}^n)$  be any distribution of teams assignments. Let  $\mathcal{G}_{n,D}$  be the class of all hidden-role games with  $n$  players and team assignment distribution  $D$ . Then the price of hidden roles of  $\mathcal{G}_{n,D}$  is exactly the largest probability assigned to any team by  $D$ , that is,

$$\text{PoHR}(\mathcal{G}_{n,D}) = \max_{t \in \{\blacktriangle, \blacktriangledown\}^n} \Pr_{t' \sim D}[t' = t].$$

The lower bound is achieved even in the presence of private communication.

*Proof.* Let  $t^*$  be the team to which  $D$  assigns the highest probability, and let  $p^*$  be that probability. Our goal is to show that the price of hidden roles is  $1/p^*$ .

(Upper bound) Team  $\blacktriangle$  assumes that the true  $\blacktriangle$ -team is exactly the team  $t^*$ . Then  $\blacktriangle$  gets utility at most a factor of  $1/p^*$  worse than the TME value of  $\text{PUBLICTEAM}(\Gamma)$ : if the assumption is correct, then  $\blacktriangle$  gets the TME value; if the assumption is incorrect,  $\blacktriangle$  gets value at least 0 thanks to the condition on  $\blacktriangle$ 's utilities in Definition 4.18.

(Lower bound) Consider the following game  $\Gamma$ . Nature first selects a team assignment  $t \sim D$  and each player privately observes its team assignment. Then, all players are simultaneously asked to announce what they believe the true team assignment is. The  $\blacktriangle$ -team wins if every  $\blacktriangle$ -player announces the true team assignment. If  $\blacktriangle$  wins,  $\blacktriangle$  gets utility 1; otherwise  $\blacktriangle$  gets utility 0.

Clearly, if teams are made public,  $\blacktriangle$  wins easily. With teams not public, suppose that we add a mediator to the game so that Theorem 4.10 applies. This cannot decrease  $\blacktriangle$ 's value. The mediator's strategy amounts to selecting what team each player should announce. Mediator strategies in which different players announce different teams are dominated. The mediator strategy in which the mediator tells every player to announce team  $t$  wins if and only if  $t$  is the true team, which happens with probability at most  $p^*$  (if  $t = t^*$ ). Thus, even the game with a mediator added has value at most  $p^*$ , completing the proof.  $\square$

This implies immediately:

**Corollary 4.20.** Let  $\mathcal{G}_{n,k}$  be the class of all hidden-role games where the number of players and adversaries are always exactly  $n$  and  $k$  respectively. The price of hidden roles in  $\mathcal{G}_{n,k}$  is exactly  $\binom{n}{k}$ .

Variant	5 Players	6 Players
No special roles ( <i>Resistance</i> )	3 / 10 = 0.3000*	1 / 3 $\approx$ 0.3333*
Merlin	2 / 3 $\approx$ 0.6667*	3 / 4 = 0.7500*
Merlin + Mordred	731 / 1782 $\approx$ 0.4102	6543 / 12464 $\approx$ 0.5250
Merlin + 2 Mordreds	5 / 18 $\approx$ 0.2778	99 / 340 $\approx$ 0.2912
Merlin + Mordred + Percival + Morgana	67 / 120 $\approx$ 0.5583	—

**Table 9:** Exact equilibrium values for 5- and 6-player Avalon. The values marked \* were also manually derived by [Christiano \(2018\)](#); we match their results. ‘—’: too large to solve.

In particular, when  $k = 1$ , the price of hidden roles is at worst  $n$ . This is in sharp contrast to the *price of communication* and *price of correlation* in ATGs, both of which can be arbitrarily large even when  $n = 3$  and  $k = 1$  ([Basilico et al., 2017](#); [Celli and Gatti, 2018](#)).

#### 4.6.2 Order of Commitment and Duality Gap

In Definition 4.8, when choosing the TME as our solution concept and defining the split-personality game, we explicitly choose that  $\blacktriangle$  should pick its strategy before  $\blacktriangledown$ —that is, the team committing to a strategy is the same one that has incomplete information about the roles. One may ask whether this choice is necessary or relevant: for example, what happens when the TME problem (4) satisfies the minimax theorem? Perhaps surprisingly, the answer to this question is that, at least with private communication, *the minimax theorem in hidden-role games only holds in “trivial” cases*, in particular, when the hidden-role game is equivalent to its public-role refinement (Definition 4.17).

**Proposition 4.21.** *Let  $\Gamma$  be any hidden-role game. Define  $\text{UVal}'_{\text{priv}}(\Gamma)$  identically to  $\text{UVal}_{\text{priv}}(\Gamma)$ , except that  $\blacktriangledown$  commits before  $\blacktriangle$ —that is, in (4), the maximization and minimization are flipped. Then  $\text{UVal}'_{\text{priv}}(\Gamma)$  is equal to the TME value of  $\text{PUBLICTEAM}(\Gamma)$  with communication—that is, the equilibrium value of the zero-sum game in which teams are public and intra-team communication is private and unlimited.*

*Proof.* It suffices to show that team  $\blacktriangle$  can always cause the teams to be revealed publicly if  $\blacktriangledown$  commits first. Let  $s$  be a long random string. All members of team  $\blacktriangle$  broadcast  $s$  publicly at the start of the game. Since  $\blacktriangledown$  commits first,  $\blacktriangledown$  cannot know or guess  $s$  if it is sufficiently long; thus, with exponentially-good probability, this completely reveals the teams publicly. Then, using the private communication channels, team  $\blacktriangle$  can play a TMECom of  $\text{PUBLICTEAM}(\Gamma)$ .  $\square$

Therefore, the choice of having  $\blacktriangle$  commit to a strategy before  $\blacktriangledown$  is forced upon us: flipping the order of commitment would ruin the point of hidden-role games.

### 4.7 Experimental Evaluation: *Avalon*

In this section, we apply Theorem 4.11 to instances of the popular hidden-role game *The Resistance: Avalon* (hereafter simply *Avalon*). We solve various versions of the game with up to six players.

A game of *Avalon* proceeds, generically speaking, as follows. There are  $n$  players,  $\lceil n/3 \rceil$  of which are randomly assigned to team  $\blacktriangledown$  and the rest to team  $\blacktriangle$ . Team  $\blacktriangledown$  is informed. Some special rules allow players observe further information; for example, *Merlin* is a  $\blacktriangle$ -player who observes the identity of the players on team  $\blacktriangledown$ , except the  $\blacktriangledown$ -player *Mordred*, and the  $\blacktriangle$ -player *Percival* knows *Merlin* and *Morgana* (who is on team  $\blacktriangledown$ ), but does not know which is which. The game proceeds in five rounds. In each round, a *leader* publicly selects a certain number of people (defined as a function of the number of players and current round number) to go on a *mission*. Players then publicly vote on whether to accept the leader’s choice. If a strict majority vote to accept, the mission begins; otherwise, leadership goes to the player to the left. If four votes pass with

no mission selected, there is no vote on the fifth mission (it automatically gets accepted). If a  $\blacktriangledown$ -player is sent on a mission, they have the chance to *fail* the mission. The goal of  $\blacktriangle$  is to have three missions pass. If *Merlin* is present,  $\blacktriangledown$  also wins by correctly guessing the identity of Merlin at the end of the game. *Avalon* is therefore parameterized by the number of players and the presence of the extra roles *Merlin*, *Mordred*, *Percival*, and *Morgana*.

*Avalon* is far too large to be written in memory: Serrino et al. (2019) calculates that 5-player *Avalon* has at least  $10^{56}$  information sets. However, in *Avalon* with  $\leq 6$  players, many simplifications can be made to the zero-sum game given by Theorem 4.11 without changing the equilibrium. These are detailed in the appendix of the full paper (Carminati et al., 2024b), but here we sketch one of them which has theoretical implications. Without loss of generality, in the zero-sum game in Theorem 4.11, the mediator completely dictates the choice of missions by telling everyone to propose the same mission and vote to accept missions, and  $\blacktriangledown$  can do nothing to stop this. Therefore, team  $\blacktriangledown$  always has symmetric information in the game: they know each others’ roles (at least when  $n \leq 6$ ), and the mediator’s recommendations to the players may as well be public. Therefore, *Avalon* is already natively without loss of generality a game with a coordinated adversary in the sense of Section 4.3.2, so the seemingly strong assumptions used in Definition 4.6 are in fact appropriate in *Avalon*. Even after our simplifications, the games are fairly large, *e.g.*, the largest instance we solve has 2.2 million infosets and 26 million terminal nodes.

Our results are summarized in Table 9. Games were solved using a CPU compute cluster machine with 64 CPUs and 480 GB RAM, using two algorithms:

1. A parallelized version of the PCFR+ algorithm (Farina et al., 2021c), a scalable no-regret learning algorithm. PCFR+ was able to find an approximate equilibrium with exploitability  $< 10^{-3}$  in less than 10 minutes in the largest game instance, and was able to complete 10,000 iterations in under two hours for each game.
2. An implementation of the simplex algorithm with exact (rational) precision, which was warmstarted using incrementally higher-precision solutions obtained from configurable finite-precision floating-point arithmetic implementation of the simplex algorithm, using an algorithm similar to that of Farina et al. (2018). This method incurred significantly higher runtimes (in the order of hours to tens of hours), but had the advantage of computing *exact* game values at equilibrium.

Table 9 shows exact game values for the instances we solved.

**Findings.** We solve *Avalon* exactly in several instances with up to six players. In the simplest instances (*Resistance* or only Merlin), Christiano (2018) previously computed equilibrium values by hand. The fact that we match those results is positive evidence of the soundness of both our equilibrium concepts and our algorithms.

Curiously, as seen in Table 9, the game values are not “nice” fractions: this suggests to us that most of the equilibrium strategies will likely be inscrutable to humans. The simplest equilibrium not previously noted by Christiano, namely Merlin + 2 Mordreds with 5 players, is scrutable, and is analyzed in detail in the appendix of the full paper (Carminati et al., 2024a).

Also curiously, having Merlin and two Mordreds (*i.e.*, having a Merlin that does not actually know anything) is not the same as having no Merlin. If it were, we would expect the value of Merlin and two Mordreds to be  $0.3 \times 2/3 = 0.2$  (due to the  $1/3$  probability of  $\blacktriangledown$  randomly guessing Merlin). But, the value is actually closer to 0.28. The discrepancy is due to the “special player” implicit correlation discussed in Section 4.1.4.

## 4.8 Conclusions and Future Research

In this paper, we have initiated the formal study of hidden-role games from a game-theoretic perspective. We build on the growing literature on ATGs to define a notion of equilibrium, and give both positive and negative results surrounding the efficient computation of these equilibria. In experiments, we completely solve real-world instances of *Avalon*. As this paper introduces a new and interesting class of games, we hope that it will be the basis of many future papers as well. We leave many interesting questions open.

1. From our results, it is not even clear that hidden-role equilibria and values can be computed in *finite* time except as given by Theorem 4.12. Is this possible? For example, is there a revelation-principle-like characterization for *public* communication that would allow us to fix the structure of the communication? We believe this question is particularly important, as humans playing hidden-role games are often restricted to communicating in public and cannot reasonably run the cryptographic protocols necessary to build private communication channels or perform secure MPC.
2. Changing the way in which communication works can have a ripple effect on the whole paper. One particular interesting change that we do not investigate is *anonymous* messaging, in which players can, publicly or privately, send messages that do not leak their own identity. How does the possibility of anonymous messaging affect the central results of this paper?
3. In this paper, we do not investigate or define hidden-role games where *both* teams have imperfect information about the team assignment. What difference would that make? In particular, is there a way to define an equilibrium concept in that setting that is “symmetric” in the sense that it does not require a seemingly-arbitrary choice of which team ought to commit first to its strategy?

## Part II

# Generalized Mechanism Design and Optimal Correlation via Zero-Sum Games

## 5 Polynomial-Time Optimal Equilibria with a Mediator

### 5.1 Introduction

Various equilibrium notions in general-sum extensive-form games are used to describe situations where the players have access to a trusted third-party *mediator*, who can communicate with the players. Depending on the power of the mediator and the form of communication, these notions include the *normal-form* (Aumann, 1974) and *extensive-form correlated equilibrium* (NFCE and EFCE) (von Stengel and Forges, 2008), the *normal-form* (Moulin and Vial, 1978) and *extensive-form* (Farina et al., 2020) *coarse-correlated equilibrium* (NFCCE and EFCCE), the *communication equilibrium* (Forges, 1986; Myerson, 1986), and the *certification equilibrium* (Forges and Koessler, 2005).

Several of these notions, in particular the EFCE and EFCCE, were defined for mainly *computational* reasons: the EFCE as a computationally-reasonable relaxation to NFCE, and the EFCCE as a computationally-reasonable relaxation of EFCE. When the goal is to compute a *single* correlated equilibrium, these relaxations are helpful: there are polynomial-time algorithms for computing an EFCE (Huang and von Stengel, 2008). However, from the perspective of computing *optimal* equilibria—that is, equilibria that maximize the expected value of a given function, such as the social welfare—even these relaxations fall short: for all of the *correlation* notions above, computing an optimal equilibrium of an extensive-form game is NP-hard (von Stengel and Forges, 2008; Farina et al., 2020).

On the other hand, notions of equilibrium involving *communication* in games have arisen. These differ from the notions of *correlation* in that the mediator can receive and remember information from the players, and therefore pass information *between* players as necessary to back up their suggestions. *Certification equilibria* (Forges and Koessler, 2005) further strengthen communication equilibria by allowing players to *prove* certain information to the mediator. To our knowledge, the computational complexity of optimal communication or certification equilibria has never been studied. We do so in this paper. The main technical result of our paper is a *polynomial-time algorithm* for computing optimal communication and certification equilibria (the latter under a certain natural condition about what messages the players can send). This stands in stark contrast to the notions of correlation discussed above.

To prove our main result, we define a general class of *mediator-augmented games*, each having polynomial size, that is sufficient to describe all of the above notions of equilibrium except the NFCE<sup>37</sup>. We also build on this main result in several ways.

1. We define the *full-certification* equilibrium, which is the special case in which players cannot lie to the mediator (but can opt out of revealing their information). In this case, the algorithm is a linear program whose size is *almost linear* in the size of the original game. As such, this special case scales extremely well compared to all of the other notions.
2. We formalize notions for incorporating *payments* in the language of our augmented game. By using

---

<sup>37</sup>We do not consider the NFCE, because it breaks our paradigm, which enforces that the mediator’s recommendation be a single action. In NFCE, the whole strategy needs to be revealed upfront. It is an open question whether it is possible to even find *one* NFCE in polynomial time, not to mention an optimal one.

payments, mediators can incentivize players to play differently than they otherwise would, possibly to the benefit of the mediator’s utility function.

3. We define an entire family of equilibria using our augmented game, that includes as special cases the communication equilibrium, certification equilibrium, NFCCE, EFCCE, and EFCE. From this perspective, we show that other notions of equilibrium, such as extensive-form correlated equilibrium, correspond to the mediator having *imperfect recall*. This shows that, at least among all these equilibrium notions, the hardness of computation is driven by the mediator’s imperfect recall. We argue that, for this reason, many stated practical applications of correlated equilibria should actually be using communication or certification equilibria instead, which are both easier to compute (in theory, at least) and better at modelling the decision-making process of a rational mediator.
4. We empirically verify the above claims via experiments on a standard set of game instances.

**Applications and related work.** Correlated and communication equilibria have various applications that have been well-documented. Here, we discuss just a few of them, as motivation for our paper. For further discussion of related work, especially relating to automated dynamic mechanism design and persuasion, see the appendix of the full paper (Zhang and Sandholm, 2022a).

*Bargaining, negotiation, and conflict resolution* (Chalamish and Kraus, 2012; Farina et al., 2019b). Two parties with asymmetric information wish to arrive at an agreement, say, the price of an item. A mediator, such as a central third-party marketplace, does not know the players’ information but can communicate with the players.

*Crowdsourcing and ridesharing* (Furuhata et al., 2013; Ma et al., 2021; Zhang et al., 2022b). A group of players each has individual goals (*e.g.*, to make money by serving customers at specific locations). The players are coordinated by a central party (*e.g.*, a ridesharing company) that has more information than any one of the players, but the players are free to ignore recommendations if they so choose.

*Persuasion in games* (Kamenica and Gentzkow, 2011; Celli et al., 2020a; Mansour et al., 2022b; Gan et al., 2022; Wu et al., 2022). The mediator (in that literature, usually “sender”) has more information than the players (“receivers”), and wishes to tell information to the receivers so as to persuade them to act in a certain way.

*Automated mechanism design* (Conitzer and Sandholm, 2002, 2004; Zhang and Conitzer, 2021; Zhang et al., 2022c; Papadimitriou et al., 2022; Zhang et al., 2021; Kephart and Conitzer, 2015, 2021). Players have private information unknown to the mediator. The mediator wishes to commit to a strategy—that is, set a mechanism—such that players are incentivized to honestly reveal their information. In fact, in the appendix of the full paper (Zhang and Sandholm, 2022a) we will see that we recover the polynomial-time Bayes-Nash randomized mechanism design algorithm of (Conitzer and Sandholm, 2002, 2004) as a special case of our main result.

Some of the above examples are often used to motivate correlated equilibria. However, when the mediator is a rational agent with the ability to remember information that it is told and pass the information between players as necessary, we will argue that communication or certification equilibrium should be the notion of choice, for both conceptual and computational reasons.

## 5.2 Preliminaries: Communication and Certification Equilibria

Here, we review definitions related to *communication equilibria*, following Forges (1986); Myerson (1986) and later related papers.

**Definition 5.1.** Let  $S$  be a space of possible *messages*. A *pure mediator strategy* is a map  $d : S^{\leq T} \rightarrow S$ , where  $S^{\leq T}$  denotes the set of sequences in  $S$  of length at most  $T$ . A *randomized mediator strategy* (hereafter simply *mediator strategy*) is a distribution over pure mediator strategies.

We will assume that the space of possible messages is large, but not exponentially so. In particular, we will assume that  $\{\perp\} \cup \mathcal{I} \cup \bigcup_h \mathcal{A}(h) \subseteq S$  (*i.e.*, messages can at least be nothing, information, or actions)<sup>38</sup> and that  $|S| \leq \text{poly}(|\mathcal{H}|)$ . The latter assumption is mostly for cleanliness in stating results: we will give algorithms that need  $S$  as an input that we wish to run in time  $\text{poly}(|\mathcal{H}|)$ .

A mediator strategy augments a game as follows. If the strategy is randomized, it first samples a pure strategy  $d$ , which is hidden from the players. At each timestep  $t$ , a player reaches a history  $h$  at which she must act, and observes the infoset  $I \ni h$ . She sends a message  $s_t \in S$  to the mediator. The mediator then sends a response  $d(s_1, \dots, s_t)$ , which depends on the message  $s_t$  as well as the messages sent by all other players prior to timestep  $t$ . Then, the player chooses her action  $a \in \mathcal{A}(h)$ . We will call the sequence of messages sent and received between the mediator and player  $i$ , the *transcript with player  $i$* . A *communication equilibrium*<sup>39</sup> is a Nash equilibrium of the game  $\Gamma$  augmented with a mediator strategy. The mediator is allowed to perform arbitrary communication with the players. In particular, the mediator is allowed to *pass information from one player to another*. Further, the players are free to send whatever messages they wish to the mediator, including false or empty messages. These two factors distinguish communication equilibria from notions of *correlated equilibria*. In Section 5.3.4 we will discuss this comparison in greater detail.

A useful property in the literature on communication equilibria is the *revelation principle* (*e.g.*, (Forges, 1986; Myerson, 1986)). Informally, the revelation principle states that any outcome achievable by an *arbitrary* strategy profile can also be achieved by a *direct* strategy profile, in which the players tell the mediator all their information and are subsequently directly told by the mediator which action to play. In order to be a communication equilibrium, the players still must not have any incentive to deviate from the protocol. That is, the equilibrium must be *robust* to all messages that a player may attempt to send to the mediator, even if *in equilibrium* the player always sends the honest message.

Forges and Koessler (2005) further introduced a form of equilibrium for Bayesian games which they called *certification equilibria*. In certification equilibria, the messages that a player may legally send are dependent on their information; as such, some messages that a player can send are *verifiable*. At each information set  $I \in \mathcal{I}$ , let  $S_I \subseteq S$  denote the set of messages that the player at infoset  $I$  may send to the mediator. We will always assume that  $I \in S_I$  and  $\perp \in S_I$  for all  $I$ . That is, all players always have the options of revealing their true information or revealing nothing.

<sup>38</sup>*A priori*, although the messages are given these names, they carry no semantic meaning. The revelation principle is used to assign natural meaning to the messages.

<sup>39</sup>Previous models of communication in games (Forges, 1986; Myerson, 1986; Forges and Koessler, 2005) usually worked with a model in which players send messages, receive messages, and play moves *simultaneously*, rather than in sequence as in the extensive-game model that we use. The simultaneous-move model is easy to recreate in extensive form: by adding further “dummy nodes” at which players learn information but only have one legal action, we can effectively re-order when players ought to communicate their information to the mediator.



### 5.3 Extensive-Form $\mathcal{S}$ -Certification Equilibria

The central notion of interest in this paper is a generalization of the notion of certification equilibria (Forges and Koessler, 2005) to extensive-form games.

**Definition 5.2.** Given an extensive-form game  $\Gamma$  and a family of valid message sets  $\mathcal{S} = \{S_I : I \in \mathcal{I}\}$ , an  *$\mathcal{S}$ -certification equilibrium* is a Nash equilibrium of the game augmented by a randomized mediator, in which each player at each information set  $I$  is restricted to sending a message  $s \in S_I$ .

The existence of  $\mathcal{S}$ -certification equilibria follows from the existence of *Nash* equilibria, which are the special case where the mediator does nothing.

We will need one extra condition on the message sets, which is known as the *nested range condition* (NRC) (Green and Laffont, 1977): if  $I \in S_{I'}$ , then  $S_I \subseteq S_{I'}$ . That is, if a player with information  $I'$  can lie by pretending to have information  $I$ , then that player can also emulate any other message she would have been able to send at  $I$ . Equivalently, the honest message  $I$  should be the *most certifiable* message that a player can send at infoset  $I$ . Our main result is the following.

**Theorem 5.3.** *Let  $\mathbf{u}_M \in \mathbb{R}^Z$  be an arbitrary utility vector for the mediator. Then there is a polynomial-time algorithm that, given a game  $\Gamma$  and a message set family  $\mathcal{S}$  satisfying the nested range condition, computes an optimal  $\mathcal{S}$ -certification equilibrium, that is, one that maximizes  $\mathbb{E}_z u_M[z]$  where the expectation is over payouts of the game under equilibrium.*

In particular, by setting  $S_I = S$  for all  $I$ , Theorem 5.3 implies that optimal communication equilibria can be computed in polynomial time.

The rest of the paper is organized as follows. First, we will prove our main theorem. Along the way, we will demonstrate a form of revelation principle for  $\mathcal{S}$ -certification equilibria. We will then discuss comparisons to other known forms of equilibrium, including the extensive-form correlated equilibrium (von Stengel and Forges, 2008), and several other natural extensions of our model. Finally, we will show experimental results that compare the computational efficiency and social welfare of various notions of equilibrium on some experimental game instances.

#### 5.3.1 Proof of Theorem 5.3: The Single-Deviator Mediator-Augmented Game

In this section, we construct a game  $\hat{\Gamma}$ , with  $n + 1$  players, that describes the game  $\Gamma$  where the mediator has been added as an explicit player. This game has similar structure to the one used by Forges (1986, Corollary 2), but, critically, has size polynomial in  $|\mathcal{H}|$ . This is due to two critical differences. First, the players are assumed to either send  $\perp$ , or send messages that mediator cannot immediately prove to be off-equilibrium. In particular, if the player's last message was  $I$  and the mediator recommended action  $a$  at  $I$ , the player must send a message  $I'$  with  $\sigma(I') = Ia$ . If this is impossible, the player must send  $\perp$ . Therefore, in particular, we will assume that  $S_I$  consists of only  $\perp$  and information sets  $I'$  at the same level as  $I$ . Second, only one player is allowed to deviate. Therefore, the strategy of the mediator is not defined in cases where two or more players deviate.

We now formalize  $\hat{\Gamma}$ . Nodes in  $\hat{\Gamma}$  will be identified by tuples  $(h, \tau, r)$  where  $h \in \mathcal{H}$  is a history in  $\Gamma$ ,  $\tau = (\tau_1, \dots, \tau_n)$  is the collection of transcripts with all players, and  $r \in \{\text{REV}, \text{REC}, \text{ACT}\}$  is a *stage marker* that denotes whether the current state is one in which a player should be *revealing information* (REV), the mediator should be *recommending a move* (REC), or the player should be *selecting an action* (ACT). The progression of  $\hat{\Gamma}$  is then defined as follows. We will use the notation  $\tau[i \cdot s]$  to denote appending message  $s$  to  $\tau_i$ .

- The root node of  $\hat{\Gamma}$  is  $(\emptyset, (\emptyset, \dots, \emptyset), \text{REV})$ .
- Nodes  $(z, \tau, \text{REV})$  for  $z \in \mathcal{Z}$  are also terminal in  $\Gamma$ . The mediator gets utility  $u_M[z]$ , where  $u$  is the mediator's utility function as in Theorem 5.3. All other players  $i$  get utility  $u_i[z]$ .
- Nodes  $(h, \tau, \text{REV})$  for non-terminal  $h$  are decision nodes for the player  $i$  who acts at  $h$ .

1. If  $i$  is chance, there is one valid transition, to  $(h, \tau, \text{ACT})$ .
  2. If some other player  $j \neq i$  has already deviated (i.e.,  $\sigma_j(h) \neq \tau_j$ ), there is one valid transition, to  $(h, \tau[i \cdot I], \text{REC})$  where  $I \ni h$ .
  3. If player  $i$  has deviated or no one has deviated, then player  $i$  observes the infoset  $I \ni h$ , and selects a legal message  $I' \in S_I \cap (\{\perp\} \cup N(\tau_i))$  to send to the mediator<sup>40</sup>. Transition to  $(h, \tau[i \cdot I'], \text{REV})$ .
- At  $(h, \tau, \text{REC})$  where  $h \in \mathcal{H}_i$ , the mediator observes the transcript  $\tau_i$  and makes a *recommendation*  $a$ . If  $\tau_i$  contains any  $\perp$  messages, then  $a = \perp$ . Otherwise,  $a$  is a legal action  $a \in \mathcal{A}(I)$ , where  $I$  is the most recent message in  $\tau_i$ . Transition to  $(h, \tau[i \cdot a], \text{ACT})$ .
  - Nodes  $(h, \tau, \text{ACT})$  for non-terminal  $h$  are decision nodes for the player  $i$  who acts at  $h$ .
    1. If  $i$  is chance, then chance samples a random action  $a \sim p(\cdot|h)$ . Transition to  $(ha, \tau, \text{REV})$ .
    2. If some other player  $j \neq i$  has already deviated, there is one valid transition, to  $(ha, \tau, \text{REC})$ , where  $a$  is the action sent by the mediator.
    3. If player  $i$  has deviated or no one has deviated, then player  $i$  observes the transcript  $\tau_i$ , and selects an action  $a' \in \mathcal{A}(h)$ . Transition to  $(ha', \tau, \text{REV})$ . The action  $a'$  need not be the recommended action.

Since at most one player can ever deviate by construction, and the length of the transcripts are fixed because turn order is common knowledge, the transcripts  $\tau$  can be identified with *sequences*  $\sigma_i$  of the deviated player, if any. We will make this identification: we will use the shorthand  $h^{\sigma_i}$  to denote the history  $(h, (\sigma_{-i}(h), \sigma_i), \text{REV})$ , and  $h^\perp$  for  $(h, \sigma(h), \text{REV})$  (i.e., no one has deviated yet). Therefore, in particular, this game has at most  $O(|\mathcal{H}||\Sigma|)$  histories.

For each non-mediator player, there is a well-defined *direct strategy*  $\hat{o}_i$  for that player: always report her true information  $I \ni h$ , and always play the action recommended by the mediator. The goal of the mediator is to *find a strategy  $\hat{x}_M$  for itself that maximizes its expected utility, subject to the constraint that each player's direct strategy is a best response*—that is, find  $\hat{x}_M$  such that  $(\hat{x}_M, \hat{o}_1, \dots, \hat{o}_n)$  is a (strong) Stackelberg equilibrium of  $\hat{\Gamma}$ .

We claim that finding a mediator strategy  $\hat{x}_M$  that is a strong Stackelberg equilibrium in  $\hat{\Gamma}$  is equivalent to finding an optimal  $\mathcal{S}$ -certification equilibrium in  $\Gamma$ . We prove this in two parts. First, we prove a version of the revelation principle for  $\mathcal{S}$ -certification equilibria.

**Definition 5.4.** An  $\mathcal{S}$ -certification equilibrium is *direct* if it satisfies the following two properties.

1. (*Mediator directness*) If the transcript  $\tau_i$  of a player  $i$  is exactly some sequence of player  $i$ , and player  $i$  sends an infoset  $I$  with  $\sigma(I) = \tau_i$ , then the mediator replies with an action  $a \in \mathcal{A}(I)$ . Otherwise<sup>41</sup>, the mediator replies  $\perp$ .
2. (*Player directness*) In equilibrium, players always send their true information  $I$ , and, upon receiving an action  $a \in \mathcal{A}(I)$ , always play that action.

**Proposition 5.5** (Revelation principle for  $\mathcal{S}$ -certification equilibria under NRC). *Assume that  $\mathcal{S}$  satisfies the nested range condition. For any  $\mathcal{S}$ -certification equilibrium, there is a realization-equivalent direct equilibrium.*

Omitted proofs can be found in the appendix of the full paper (Zhang and Sandholm, 2022a). Since direct mediator strategies are exactly the mediator strategies in  $\hat{\Gamma}$ , and the player strategies are only limited versions of what they are allowed to do in  $\mathcal{S}$ -certification equilibrium, this implies that, for any  $\mathcal{S}$ -certification equilibrium, there is a mediator strategy  $\hat{x}_M$  in  $\hat{\Gamma}$  such that  $(\hat{x}_M, \hat{o}_1, \dots, \hat{o}_n)$  is a Stackelberg equilibrium. We will also need the converse of this statement.

<sup>40</sup>If  $\tau_i$  contains any  $\perp$  messages, then we take  $N(\tau_i) = \emptyset$

<sup>41</sup>This condition is necessary because, if the mediator does not know what infoset the player is in, the mediator may not be able to send the player a valid action, because action sets may differ by infoset.

**Proposition 5.6.** Let  $\hat{\mathbf{x}}_M$  be a strategy for the mediator in  $\hat{\Gamma}$  such that, in the strategy profile  $(\hat{\mathbf{x}}_M, \hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_n)$ , every  $\hat{\mathbf{o}}_i$  for  $i \neq M$  is a best response. Then there is an direct  $\mathcal{S}$ -certification equilibrium that is realization-equivalent to  $(\hat{\mathbf{x}}_M, \hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_n)$ .

Therefore, we have shown that the mediator strategies  $\hat{\mathbf{x}}_M$  in  $\hat{\Gamma}$  for which  $(\hat{\mathbf{x}}_M, \hat{\mathbf{o}}_1, \dots, \hat{\mathbf{o}}_n)$  is a Stackelberg equilibrium in  $\hat{\Gamma}$  correspond exactly to optimal  $\mathcal{S}$ -certification equilibria of  $\Gamma$ . Such a Stackelberg equilibrium can be found by solving the following program:

$$\begin{aligned} \max_{\hat{\mathbf{x}}_M \in \text{co } \hat{\mathcal{X}}_M} \quad & \sum_{\hat{z} \in \hat{\mathcal{Z}}} \hat{\mathbf{x}}_M[\hat{z}] \hat{\mathbf{u}}_M[\hat{z}] \hat{p}(\hat{z}) \prod_{i \in [n]} \hat{\mathbf{o}}_i[\hat{z}] \\ \text{s.t.} \quad & \max_{\hat{\mathbf{x}}'_j \in \text{co } \hat{\mathcal{X}}_j} \sum_{\hat{z} \in \hat{\mathcal{Z}}} \hat{\mathbf{x}}_M[\hat{z}] \hat{\mathbf{u}}_i[\hat{z}] \hat{p}(\hat{z}) (\hat{\mathbf{x}}'_j[\hat{z}] - \hat{\mathbf{o}}_j[\hat{z}]) \prod_{i \neq j} \hat{\mathbf{o}}_i[\hat{z}] \leq 0 \quad \forall j \in [n] \end{aligned} \quad (5)$$

where  $\text{co } \hat{\mathcal{X}}_i$  is the sequence-form mixed strategy space (Koller et al., 1994) of player  $i$  in  $\hat{\Gamma}$ .

The only variables in the program are  $\hat{\mathbf{x}}_i$  for each player  $i$  and the mediator. In particular, the direct strategies  $\hat{\mathbf{x}}_i^*$  are constants. Therefore, the objective is a linear function, and the inner maximization constraints are bilinear in  $\hat{\mathbf{x}}_M$  and  $\hat{\mathbf{x}}_j$ . Therefore, this program can be converted to a linear program by dualizing the inner optimizations. The result is a linear program of size  $O(n|\hat{\mathcal{H}}|) = O(n|\mathcal{H}||\Sigma|)$ . We have thus proved Theorem 5.3.

### 5.3.2 Extensions and Special Cases

In this section, we describe several extensions and interesting special cases of our main result.

**Full-certification equilibria.** One particular special case of  $\mathcal{S}$ -certification equilibria which is particularly useful. We define a *full-certification equilibrium* as an  $\mathcal{S}$ -certification equilibrium where  $S_I = \{\perp, I\}$ . Intuitively, this means that players cannot *lie* to the mediator, but they may *withhold* information. We will call such an equilibrium *full-certification*. Removing valid messages from the players only reduces their ability to deviate and thus increases the space of possible equilibrium strategies. As such, the full-certification equilibria are the largest class of  $\mathcal{S}$ -certification equilibria.

For full-certification equilibria, the size of game  $\hat{\Gamma}$  reduces dramatically. Indeed, in all histories  $h^{Ia}$  of  $\hat{\Gamma}$ , we must have  $I \preceq h$ . Therefore, we have  $|\hat{\mathcal{H}}| \leq |\mathcal{H}|BD$  where  $B$  is the maximum branching factor and  $D$  is the depth of the game tree, *i.e.*, the size of  $\hat{\Gamma}$  goes from essentially quadratic to essentially quasilinear in  $|\mathcal{H}|$ . The mediator's decision points in  $\hat{\Gamma}$  for a full-certification equilibrium are the *trigger histories* used by Zhang et al. (2022b) in their analysis of various notions of correlated equilibria. Later, we will draw further connections between full certification and correlation.

**Changing the mediator's information.** In certain cases, the mediator, in addition to messages that it is sent by the players, also has its own observations about the world. These are trivial to incorporate into our model: simply change the information partition of the mediator in  $\hat{\Gamma}$  as needed. Alternatively, one can imagine adding a “player”, with no rewards (hence no incentive to deviate), whose sole purpose is to observe information and pass it to the mediator. For purposes of keeping the game small, it is easier to adopt the former method. To this end, consider any refinement partition  $\mathcal{M}$  of the mediator infosets in  $\hat{\Gamma}$ , and consider the game  $\hat{\Gamma}^{\mathcal{M}}$  created by replacing the mediator's information partition in  $\hat{\Gamma}$  with  $\mathcal{M}$ . Then we make the following definition.

**Definition 5.7.** An  $(\mathcal{S}, \mathcal{M})$ -certification equilibrium of  $\Gamma$  is a mediator strategy  $\hat{\mathbf{x}}_M$  in  $\hat{\Gamma}^{\mathcal{M}}$  such that, in the strategy profile  $(\hat{\mathbf{x}}_M, \hat{\mathbf{x}}_1^*, \dots, \hat{\mathbf{x}}_n^*)$ , every  $\mathbf{o}_i$  for  $i \neq M$  is a best response.

$(\mathcal{S}, \mathcal{M})$ -certification equilibria may not exist: indeed, if  $\mathcal{M}$  is coarser than the mediator's original information partition in  $\hat{\Gamma}$ , then the mediator may not have enough information to provide good recommendations under the restrictions of  $\hat{\Gamma}$ . This can be remedied by allowing payments (see the appendix of the full paper (Zhang and Sandholm, 2022a)), or by making the assumption that the mediator *at least* knows the transcript of the player to whom she is making any nontrivial recommendation:

**Definition 5.8.** A mediator partition  $\mathcal{M}$  is *direct* if, at every mediator decision point  $(h, \tau, \text{REC})$ , so long as  $|\mathcal{A}(h)| > 1$ , the mediator knows the transcript of the player acting at  $h$ .  $\mathcal{M}$  is *strongly direct* if the mediator also observes the transcript when  $|\mathcal{A}(h)| = 1$ .

The condition  $|\mathcal{A}(h)| > 1$  in the definition allows the mediator to possibly *not* observe the full information of a player if she does not need to make a nontrivial recommendation to that player. In particular, this allows players to sometimes have information that they only partially reveal to the mediator, so long as the player does not immediately need to act on such information.

**Coarseness.** In literature on correlation, *coarseness* refers to the restriction that a player must obey any recommendation that she receives (but may choose to deviate by not requesting a recommendation and instead playing any other action). *Normal-form coarseness* further adds the restriction that players can only choose to deviate at the start of the game—the mediator essentially takes over and plays the game on behalf of non-deviating players. These notions can easily be expressed in terms of our augmented game, therefore also allowing us to express coarse versions of our equilibrium notions as augmented games.

### 5.3.3 The Gap between Polynomial and Not Polynomial

If players cannot send messages to the mediator at all, and the mediator has no other way of gaining any information, we recover the notion of *autonomous correlated equilibrium (ACE)*. It is NP-hard to compute optimal ACE, even in Bayesian games (see *e.g.*, von Stengel and Forges (2008)).

When  $\mathcal{M}$  is direct and perfect recall, computing an optimal *direct*  $(\mathcal{S}, \mathcal{M})$ -certification equilibrium can be done in polynomial time using our framework. When  $\mathcal{S}$  obeys NRC and  $\mathcal{M}$  satisfies a stronger condition<sup>42</sup>, the proof of the revelation principle (Propositions 5.5 and 5.6) works, and the resulting equilibrium is guaranteed to be optimal over all possible equilibria including those that may not be direct.

If NRC does not hold, one can still solve the program (5), and the solution is still guaranteed to be an optimal *direct* equilibrium by Proposition 5.6. However, it is not guaranteed to be optimal over all possible communication structures. Indeed, Green and Laffont (1977, Theorem 1) give an instance in which, without NRC, there can be an outcome distribution that is not implementable by a direct mediator. Our program cannot find such an outcome distribution. The counterexample does not preclude the possibility of efficient algorithms for finding optimal certification equilibria in more general cases, but does give intuition for why NRC is crucial to our construction.

We could also consider changing the mediator’s information partition so that the mediator does not have perfect recall. This transformation allows us to recover notions of *correlation* in games. Indeed, if we start from the *full-certification* equilibrium and only allow the mediator to remember the transcript with the player she is currently talking to, we recover EFCE. Adding coarseness similarly recovers EFCCE and NFCCE. In this setting, the inability to represent the strategy space of an imperfect-recall player may result in the loss of efficient algorithms.

---

<sup>42</sup>Roughly speaking, this condition is that players should not be able to cause the mediator to gain information apart from their own messages by sending messages. It holds for all notions we discuss in this paper. Formalizing the general case is beyond the scope of this paper.

		ex ante	when can players deviate?	
			coarse	ex interim
			private coarse comm	not coarse
				private comm
mediator remembers only current player's transcript	lying possible	NFCCE (Moulin and Vial, 1978)	EFCCE (Farina et al., 2020)	EFCE (von Stengel and Forges, 2008)
	withholding only			
mediator information advantage		Bayes NFCCE (Celli et al., 2020a)	Bayes EFCCE	Bayes EFCE
mediator perfect recall	lying possible	("mediated" (Monderer and Tenenholz, 2009))	NF coarse full-cert	comm (Forges, 1986; Myerson, 1986)
	withholding only		coarse full-cert	full-cert (Forges and Koessler, 2005)
	mediator information advantage	Bayes PI-NFCCE	Bayes PI-EFCCE	Bayes PI-EFCE

**Table 10:** A whole family of equilibria. See Section 5.3.4 for an explanation of the terms used in the table. NF, EF, and PI stand for normal-form, extensive-form, and perfect-information respectively.

### 5.3.4 A Family of Equilibria

By varying

1. what the mediator observes,
2. whether the mediator has perfect recall,
3. whether the players can lie or only withhold information, and
4. when and how players can deviate from the mediator's recommended actions,

we can use our framework to define a family consisting of 16 conceptually different equilibrium notions. More can be generated by considering other variations in this design space, but we focus on the extreme cases in the table. Some of these were already defined in the literature; the remaining names are ours. The result is Table 10. An inclusion diagram for these notions can be found in Figure 11.

In the table, *ex ante* means that players have only a binary choice between deviating (in which case they can play whatever they want) and playing (in which case they must always be direct and obey recommendations). With *ex ante* deviations, it does not matter whether lying is allowed because we can never get to that stage: either the player deviates immediately and never communicates with the mediator, or the player is direct. If the mediator only remembers the current active player's information, and players cannot lie, withholding and coarsely deviating are the same.

*Mediator information advantage* means that the mediator always learns the infoset of the current active player, and therefore requires no messages from the players. This is equivalent to forcing players to truthfully report information. A mediator with information advantage may still not have perfect information—for example, it will not know whether a player (or nature) has played an action until some other player observes the action. In this setting, the mediator may also have extra private information (known to none of the players), leading to the setting of Bayesian persuasion (Kamenica and Gentzkow, 2011). In extensive-form games, there are two different reasonable notions of persuasion: one that stems from extending *correlated* equilibria, and one that stems from extending *communication* equilibria. The distinction is that, in the former, the mediator has imperfect recall. For a more in-depth discussion of Bayesian persuasion, see the appendix of the full paper (Zhang and Sandholm, 2022a).

Our framework allows optimal equilibria for all notions in the table to be computed. For perfect-recall mediators, this is possible in polynomial time via the sequence form; for imperfect-recall mediators, the problem is NP-hard, in general, but—as we will elaborate on later—the *team belief DAG* of Zhang et al. (2023b) can be used to recover fixed-parameter algorithms.



# 6 Optimal Correlated Equilibria in General-Sum Games: Fixed-Parameter Algorithms, Hardness, and Two-Sided Column-Generation

## 6.1 Introduction

In this section, we will study the problem of computing *optimal correlated equilibria*, in particular, how this computational problem fits into the framework introduced in the previous section.

Our focus is on computing *optimal* NFCCEs, EFCCEs, and EFCEs, which are the equilibria that maximize a given linear objective function. Computing optimal correlated equilibria, in any of these notions, is NP-hard in the size of the game tree, even in two-player games with chance nodes, or three-player games without chance nodes (von Stengel and Forges, 2008). Some special cases are known to be solvable efficiently. von Stengel and Forges (2008) show that in two-player games without chance moves, optimal equilibria in all three equilibrium notions can be computed in polynomial time. More recently, Farina and Sandholm (2020) extend the positive result to so-called *triangle-free games*, which strictly include all two-player games with public chance actions.

The problem of computing *one* EFCE (and, therefore, one NFCCE/EFCCE) can be solved in polynomial time in the size of the game tree (Huang and von Stengel, 2008) via a variation of the *Ellipsoid Against Hope* algorithm (Papadimitriou and Roughgarden, 2008; Jiang and Leyton-Brown, 2015). Moreover, there exist decentralized no-regret learning dynamics guaranteeing that the empirical frequency of play after  $T$  rounds is an  $O(1/\sqrt{T})$ -approximate EFCE with high probability, and an EFCE almost surely in the limit (Celli et al., 2020b; Farina et al., 2021b). Using regret minimizers to play large multi-player games has already led to superhuman practical performance in multi-player poker (Brown and Sandholm, 2019b). As stated above, however, computing optimal equilibria is much harder.

**Contributions and paper structure.** This paper makes a number of contributions related to the computation of *optimal* (i.e., one that maximizes a given linear objective function, such as social welfare or any weighted sum of expected player utilities) NFCCE, EFCCE, and EFCE in general multi-player general-sum extensive-form games. At a high level, we distinguish between *conceptual*, *complexity-theoretic*, and *algorithmic* contributions.

- *Conceptual contributions.* At the conceptual level, we show that the problem of computing an optimal NFCCE, EFCCE, and EFCE, can be converted into the problem of computing an optimal strategy for a player in a suitably-constructed game. The equivalent game, which we call a *mediator-augmented game*, explicitly captures the decision problem that each player would face if the correlation device were an explicit player in the game, called the *mediator*. The action space of the mediator depends on the solution concept being analyzed: NFCCE, EFCCE, or EFCE.

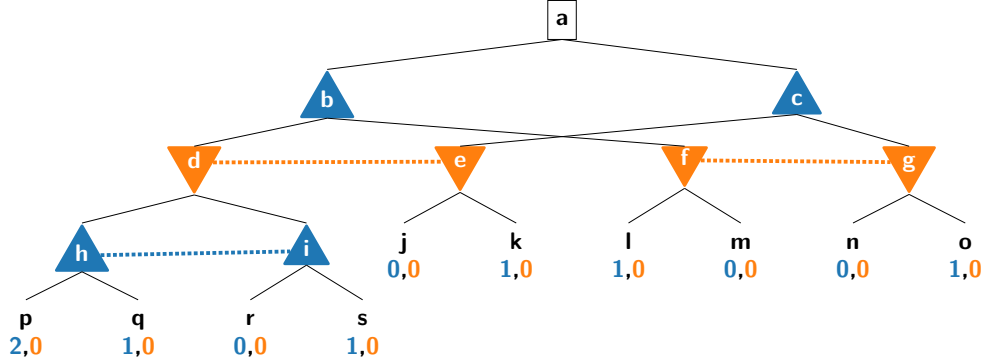
While the mediator-augmented formalism greatly simplifies the treatment—providing what we hope will be an important conceptual framework for further analysis of these solution concepts—this game reformulation preserves the computational aspects of computing an optimal equilibrium, including their hardness aspects. Indeed, a key point regarding the mediator-augmented game is that the mediator faces *imperfect recall*. This is because the mediator cannot leak information across the players, so the mediator has to forget what it has observed about the other players when making a recommendation to a given player. Otherwise, the mediator’s recommendations would not form a correlated profile at all, much less any equilibrium.

Optimizing for the strategy of an imperfect-recall player (here, the mediator) is known to be hard (Koller and Megiddo, 1992; Chu and Halpern, 2001). To tackle the issue, in our paper we study effective extended formulations (in the mathematical programming sense, e.g., (Conforti et al., 2010)) of the decision space of the mediator, by removing the imperfect recall at the expense of a (worst-case exponential) increase in the number of decision points for the mediator player.

- *Complexity-theoretic contributions.* We then proceed to show how certain recent results regarding parameterized complexity of imperfect-recall decision problems can be applied to the mediator-augmented game. A critical technical step in applying those results lies in characterizing the complexity of the *public states* of the decision problem faced by the mediator in the mediator-augmented game as a function of the original (not mediator-augmented) input game. Specifically, we give bounds on the size of the public states of mediator-augmented games for each of the solution concepts as a function of the depth  $d$ , the maximum branching factor  $b$ , and a suitably-defined *information-complexity*  $k$  of the input game that is independent of the solution concept. However, our overall complexity bounds are different depending on the solution concept: the bound for NFCCE in particular does not depend exponentially on the depth of the game, whereas the bounds for EFCCE and EFCE do. We show that this difference is inherent, therefore contributing new complexity-theoretic separations between the solution concepts.
  - i. We show that an optimal EFCE in an extensive-form game can be computed by solving a linear program of size  $O^*((bd)^k)$ , where the notation  $O^*$  suppresses factors polynomial in the size of the game (Theorem 6.15). For optimal EFCCE and optimal NFCCE, we establish bounds of  $O^*((b+d-1)^k)$  and  $O^*((b+1)^k)$ , respectively.
  - ii. In games with *public player actions*, we show that the bounds for NFCCE and EFCCE can be further improved to  $O^*(3^k)$  and  $O^*(d^k)$ , respectively (Theorem 6.17). We show that the bound for EFCE *cannot* be improved in this manner.
  - iii. In *two-player* games with *public chance actions*, our algorithm runs in polynomial time (Theorem 6.19) for all three solution concepts. The problem in this setting had already been shown to be solvable in polynomial time using a different technique by Farina and Sandholm (2020); we match their results and discuss the relationship between our algorithm and theirs in Section 6.6.1.
  - iv. We show that the gap between the NFCCE bound and the EFCCE and EFCE bounds is fundamental. Matching the bound for NFCCE—in particular, removing the dependence on  $d$ —is impossible for EFCCE and EFCE under standard complexity assumptions, demonstrating a *fundamental complexity-theoretic gap* for coarse correlation between normal and extensive form (Theorem 6.22).
- *Algorithmic contributions.* We propose two main algorithms for computing optimal correlated equilibria in all three solution concepts.
  - i. We operationalize the positive complexity results established above (Theorems 6.15, 6.17 and 6.19) via Algorithm **CorrelationDAG**. It computes an optimal strategy for the mediator in the mediator-augmented game via linear programming. At its core, the algorithm is based on the idea that the imperfect-recall strategy space of the mediator is the projection of the set of flows in a suitable high-dimensional directed acyclic graph (DAG), called the *team belief DAG* (Zhang et al., 2023b). To our knowledge, this characterization of the complicated polytope of feasible correlated equilibria as the projection of a simpler set of flows in a higher dimension is the first example of an extended formulation (in the mathematical programming sense, *e.g.*, (Conforti et al., 2010)) for these solution concepts.
 

One cannot directly apply the fixed-parameter results of Zhang et al. (2023b), as that would result in a worse bound. Instead, the above results are proven by carefully analyzing the size of the resulting construction with the special structure of the mediator-augmented games in mind.
  - ii. We propose a new practical approach to computing optimal correlated equilibria which we call *two-sided column generation* (deferred to full paper (Zhang et al., 2024c)). We start by deriving an LP formulation based on the strategy polytope of von Stengel and Forges (2008) and on the notion of *semi-randomized correlation plan* introduced by Farina et al. (2021a) in the context of team games. In the latter of those two prior approaches, one player is chosen to play a normal-form strategy and the other plays a mixed (sequence-form) strategy. Our approach improves upon this by allowing the master LP to *select* which player is chosen to play the mixed strategy, thereby increasing the space of correlation plans that can be represented for any given support, and leading to a tighter master problem. In practice, we find that this change yields a speed improvement over





**Figure 12:** An example game, between two players  $\blacktriangle$  ( $P1$ ) and  $\blacktriangledown$  ( $P2$ ). The root node is a chance node, at which chance moves uniformly at random. Dotted lines connect nodes in the same information set. Bold lowercase letters are the names of nodes. We will refer to infosets by naming all the nodes within them; for example,  $\mathbf{b}$  and  $\mathbf{de}$  are infosets. At terminal nodes, the utility of  $\blacktriangle$  is listed below the name of the node.  $\blacktriangledown$  has utility zero at every terminal node, and in this game the only role of  $\blacktriangledown$  is to incentivize  $\blacktriangle$  to act in a certain way.

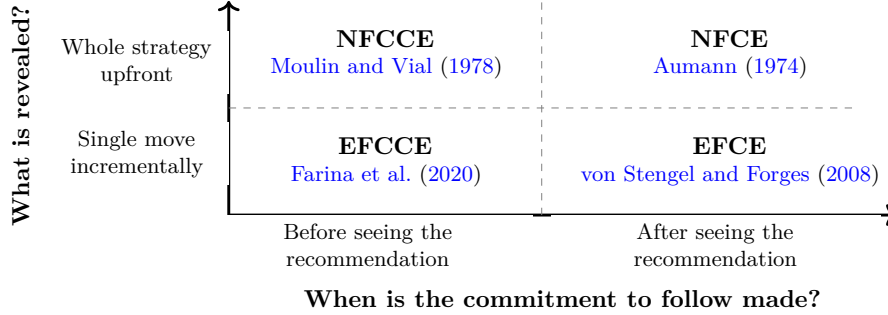
the algorithm of Farina et al. (2021a) in almost all of the games tested, and this speed improvement can be greater than two orders of magnitude.

Our two solving techniques are complementary: where the parameter  $k$  is small, writing out the DAG is superior; where it is large, the two-sided column generation is faster and more frugal in its memory usage. Furthermore, the value of  $k$  can be easily computed, enabling an efficient choice between these two approaches. In experiments (deferred to full paper (Zhang et al., 2024c)), we demonstrate state-of-the-art practical performance compared to prior state-of-the-art techniques with at least one, and sometimes both, of our techniques. We also introduce two new benchmark games: a 2-vs-1 adversarial team game we call the *tricks game* which is the trick-taking (endgame) phase of the card game bridge, and the *ride-sharing game* in which two drivers seek to earn points by serving requests across a road network modeled as an undirected graph. In the tricks game, we demonstrate empirically that, even for small endgames with only three cards per player remaining, relaxing the game to be perfect information—as so-called *double dummy* bridge endgame solvers do (e.g., (Ginsberg, 1999))—causes incorrect solutions and game values to be generated, demonstrating the need for imperfect-information game analysis.

## 6.2 Preliminaries: Correlated Equilibria in Games

Most notions of correlated equilibria in extensive-form games, including *normal-form coarse correlated equilibrium* (NFCCE), *extensive-form coarse correlated equilibrium* (EFCCE), and *extensive-form correlated equilibrium* (EFCE), can be thought of as correlated strategies of play that can be *enforced by a mediator*. The mediator first computes and publicly announces a correlated profile  $\xi$ . Then, *privately*, the mediator selects a profile  $x \sim \xi$ . Then, whenever a player  $i$  reaches an infoset  $I$ , the mediator gives a *recommendation* that  $i$  play  $x_i(I)$ . The player may also choose to *deviate*, in which case they do not need to follow the recommendations of the mediator, but the mediator also no longer *gives* recommendations for the remainder of the game. The different notions of correlation are separated by what types of deviations are allowed (see also Figure 13).

- In NFCCE, a player may only deviate at the very beginning of the game. If she chooses not to deviate, she must follow all mediator recommendations for the whole game.
- In EFCCE, a player may deviate at each of her infosets *before* seeing a recommendation. However, if she chooses not to deviate, she must play the recommended action.
- In EFCE, a player may deviate at each of her infosets *after* seeing a recommendation, by instead playing a different action.



**Figure 13:** Comparison of different notions of correlation in extensive-form games.

The fourth notion of equilibrium, called *normal-form correlated equilibrium* (NFCE), is often known as simply the *correlated equilibrium*. In NFCE, the mediator tells each player her entire pure strategy  $x_i$  at the start of the game, at which point the player may choose to deviate. It is known computing optimal NFCEs is NP-hard even in two-player games without chance nodes (unlike for the three notions we study in this paper) (von Stengel and Forges, 2008), making it a distinctly difficult problem that is out of the scope of this paper. Thus, throughout this paper, we use “*correlated equilibrium*” to generically refer to any of the three notions of correlated equilibrium that we investigate.

**Triggers.** To formalize these notions, we use the language of *deviations* introduced by Gordon et al. (2008). Each deviation consists of a *trigger* and a *continuation strategy*, which specifies the behaviour of the player when they decide to deviate from the mediator’s recommendation. The trigger determines the point of the game in which the deviating player stops following the recommendation to start playing as prescribed by the continuation strategy. Each of the solution concepts that we consider has a different set of triggers. In an NFCCE each player is allowed to deviate only at the beginning of the interaction, before any recommendation is observed. Therefore, each player  $i$  will have the empty sequence  $\emptyset_i$  as their trigger. In an EFCCE triggers are the information sets of the game, while in an EFCE players may get triggered after observing a specific action recommendation at a specific information set of the game.

**Definition 6.1.** A *trigger*  $\tau$  is:

- for NFCCE, the empty sequence  $\emptyset_i$  for some player  $i \in [n]$ ;
- for EFCCE, an infoset; and
- for EFCE, a sequence.

Given a solution concept  $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$ , we denote by  $\mathcal{T}^c$  the set of all triggers for that concept, and  $\mathcal{T}_i^c$  the set of all triggers of player  $i$ . Given a trigger  $\tau \in \mathcal{T}^c$ , we use  $\bar{\tau}$  to denote where  $\tau$  can be activated. That is,  $\bar{\tau} = I$  if  $\tau = Ia$  is a non-root sequence, or else  $\bar{\tau} = \tau$ . We must make this distinction because EFCE triggers are activated not by reaching a part of a game tree, but by receiving a recommendation  $a$  after reaching a part of the game tree. We use  $\Sigma_i^{\bar{\tau}}$  to denote the set of all sequences  $\sigma \succeq \bar{\tau}$  of player  $i$ .

A (pure) *continuation*  $\mathbf{x}'_i \in \{0, 1\}^{\Sigma_i^{\bar{\tau}}}$  following a trigger  $\tau$  of player  $i$  is a pure strategy defined on all infosets  $I \succeq \bar{\tau}$ . In sequence form,  $\mathbf{x}'_i$  is indexed by sequences  $\sigma \succeq \bar{\tau}$ , and  $\mathbf{x}'_i[\sigma] = 1$  if the player plays all actions on the path from  $\bar{\tau}$  to  $\sigma$ . Mixed continuation strategies are defined analogously.

**Deviations.** A pair  $(\tau, \mathbf{x}'_i)$ , consisting of a trigger  $\tau$  of player  $i$  and a pure continuation  $\mathbf{x}'_i$  following  $\bar{\tau}$ , defines a *deviation*  $\phi^{(\tau, \mathbf{x}'_i)} : \mathcal{X}_i \rightarrow \mathcal{X}_i$  in the following manner:  $\phi^{(\tau, \mathbf{x}'_i)}(\mathbf{x})$  is the pure strategy that plays according to the original strategy  $\mathbf{x}_i$  unless it prescribes  $\tau$ , in which case it replaces it strategy with the continuation  $\mathbf{x}'_i$  wherever the latter is defined. Formally,

$$\phi^{(\tau, \mathbf{x}'_i)}(\mathbf{x})[\sigma] := \begin{cases} \mathbf{x}'_i[\sigma] & \text{if } I \succeq \bar{\tau} \text{ and } x_i[\tau] = 1 \\ \mathbf{x}_i[\sigma] & \text{otherwise} \end{cases}$$

**Definition 6.2.** Given a correlated profile  $\mu$ , a deviation  $\phi$  of a player  $i$  is *profitable* if the deviating player improves its expected utility:  $\mathbb{E}_{\mathbf{x} \sim \mu} u_i(\phi(\mathbf{x}_i), \mathbf{x}_{-i}) > \mathbb{E}_{\mathbf{x} \sim \mu} u_i(\mathbf{x})$ .

**Definition 6.3.** NFCCEs, EFCCEs, and EFCEs are correlated profiles  $\mu$  that have no profitable deviations of their respective types.

Here, in deciding whether to deviate, the players have common knowledge of the correlated profile  $\mu$  from which their recommendations are drawn.

Given an objective function  $g : \mathcal{Z} \rightarrow \mathbb{R}$ , we say that an equilibrium  $\mu$  is *optimal* with respect to an objective  $g : \mathcal{Z} \rightarrow \mathbb{R}$  if  $\mu$  maximizes the expected objective value  $\mathbb{E}_{\mathbf{x} \sim \mu, z \sim \mathbf{x}} g(z)$  among all equilibria of the same notion.

**Remark 6.4.** The number of triggers available to a given player will play a fundamental role in the complexity of computing a solution according to each of the three solution concepts. In particular, for NFCCE, each player has only one trigger ( $\emptyset_i$ ), whereas for EFCCE and EFCE, the number of triggers for each player depends on the depth of the game. We will see in Section 6.5 that this difference results in a fundamental gap: under reasonable assumptions, an optimal NFCCE can be computed faster than an optimal EFCCE or an optimal EFCE.

### 6.3 Example of Solution Concepts

In this section, we give an example that illustrates the difference between NFCCE, EFCCE, and EFCE.

Consider the extensive-form game in Figure 12.

**Example 6.5.** As an example, consider the example game of Figure 12. The game has two players ( $n = 2$ ), whose nodes are pictorially marked with  $\blacktriangle$  for Player 1 and  $\blacktriangledown$  for Player 2 respectively, and 19 nodes (denoted  $\mathbf{a}$  through  $\mathbf{s}$ ), of which nine ( $\mathbf{a}$  through  $\mathbf{i}$ ) are nonterminal. The root node is a chance node, at which the chance player moves uniformly at random. Being the only chance node, it follows that  $\mathcal{H}_0 = \{\mathbf{a}\}$ . Player 1 ( $\blacktriangle$ ) observes the outcome of the chance node, and can pick between a left or a right action. Player 2 ( $\blacktriangledown$ ) however does not observe the outcome of the chance node; rather, the player only observes the choice of Player 1. This imperfect knowledge of the state is encoded by the information partition  $\mathcal{I}_2$  of Player 2, which contains the two information sets  $\{\{\mathbf{d}, \mathbf{e}\}, \{\mathbf{f}, \mathbf{g}\}\}$ , denoted in the figure with dotted lines connecting the nodes in the same information set. If the game hits state  $\mathbf{d}$ , then Player 1 ( $\blacktriangle$ ) gets to play a second move. However, Player 1 will not observe the action chosen by Player 2 at  $\mathbf{d}$ ; this is captured again by the information set  $\{\mathbf{h}, \mathbf{i}\}$ . Nodes  $\mathbf{b}$  and  $\mathbf{c}$  do not bear any uncertainty, and are therefore singleton elements in their corresponding information sets. In summary, the information partitions of the players are  $\mathcal{I}_1 = \{\{\mathbf{b}\}, \{\mathbf{c}\}, \{\mathbf{h}, \mathbf{i}\}\}$  and  $\mathcal{I}_2 = \{\{\mathbf{d}, \mathbf{e}\}, \{\mathbf{f}, \mathbf{g}\}\}$ . At terminal nodes, the payoffs for  $\blacktriangle, \blacktriangledown$  are listed below the node.  $\blacktriangledown$  has utility zero at every terminal node.

This game represents a signalling game between two players,  $\blacktriangle$  and  $\blacktriangledown$ .  $\blacktriangledown$  has no rewards and will therefore never have incentives to deviate from recommendations.  $\blacktriangle$  scores a point if  $\blacktriangledown$  plays the same action as chance played at the root, but chance's action is only privately revealed to  $\blacktriangle$ , so  $\blacktriangledown$  relies on  $\blacktriangle$  to signal the chance action through  $\blacktriangle$ 's own action.  $\blacktriangle$  also has the opportunity to receive a bonus point for guessing  $\blacktriangledown$ 's action in case  $\mathbf{d}$  is reached.

We will refer to the pure profiles in this game using the notation  $\boxed{\text{bcdfh}}$ , where the letters indicate which actions were played at the respective infosets containing those nodes. For example,  $\boxed{\text{LRLRL}}$  means that  $\blacktriangle$  plays left at **b**, right at **c**, and left at infoset **hi**; while  $\blacktriangledown$  plays left at **de** and right at **fg**—in particular,  $\blacktriangle$  copies chance, and  $\blacktriangledown$  copies  $\blacktriangle$ . If  $\blacktriangle$  plays right at **b**, we leave  $\blacktriangle$ 's action at **hi** unspecified since it is irrelevant; for example,  $\boxed{\text{RLRL}}$  is a valid pure strategy.

We make the following observations about our example game.

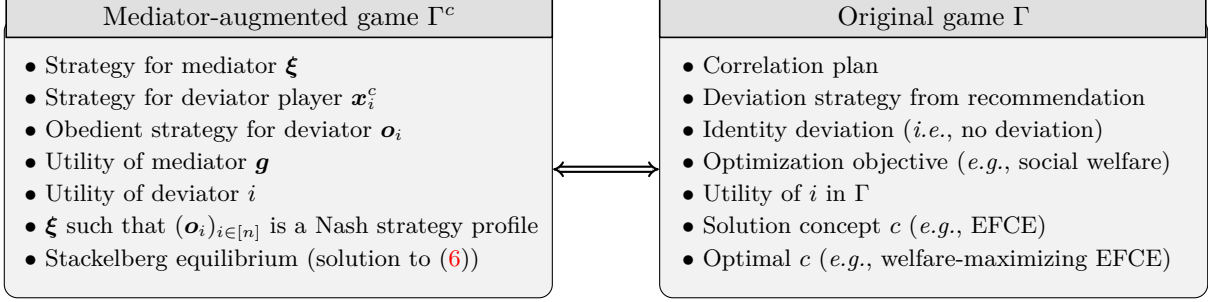
- The correlated profile  $\mu_1 := \frac{1}{2}\boxed{\text{LRLRR}} + \frac{1}{2}\boxed{\text{RLRL}}$  is an NFCCE:  $\blacktriangle$  is getting utility 1, which is larger than any utility it can get by unilaterally deviating without seeing any recommendations: since  $\blacktriangledown$ 's marginal strategy is uniform random, a best unilateral deviation for  $\blacktriangle$  is to always play left, securing expected utility  $3/4$ . However,  $\mu_1$  is not an EFCCE, because  $\blacktriangle$  can profitably deviate at trigger **hi** by playing left instead of right. This deviation cannot be expressed as an NFCCE deviation, because it requires  $\blacktriangle$  to follow recommendations at **b** and **c**.
- The correlated profile  $\mu_2 := \frac{1}{2}\boxed{\text{LRLRL}} + \frac{1}{2}\boxed{\text{RRRR}}$  is an EFCCE.  $\blacktriangle$  still gets total expected utility 1.  $\blacktriangle$  is already getting the optimal utility at **c** and **hi**; and at **b**,  $\blacktriangle$  is currently getting a conditional utility of 1, and she cannot improve upon this without seeing the recommendation at **b**. However,  $\mu_2$  is not an EFCE, because  $\blacktriangle$  can profitably deviate upon being recommended to play right at **b** by instead playing left at **b** and right at **hi**. This deviation cannot be expressed as an EFCCE deviation, because, in the deviation,  $\blacktriangle$  conditions her action at infoset **hi** on the recommendation that she received at **b**.
- The pure profile  $\boxed{\text{LRLRL}}$  is an EFCE (in fact, being uncorrelated, it is a Nash equilibrium).

## 6.4 Unifying Correlated Solution Concepts via Mediator-Augmented Games

As mentioned in the previous section and summarized in Figure 13, different correlated solution concepts for extensive-form games differ in what the mediator (correlation device) reveals to the players, and whether the players' choices to commit to follow the recommended behavior happen before or after observing the recommendation. These differences not only materialize in different equilibrium sets, but—as we will show later in this paper—also in complexity barriers that separate the solution concepts. Consequently, a unified treatment of these solution concepts needs to be approached with care.

In this section, we define augmented games in which the mediator is made explicit, which will be pivotal to our main results. Prior to presenting a precise formalization of the notion of augmented game, we provide some intuition about how the game is constructed, using the illustrative example in Figure 12. During this phase, our primary objective is to provide a straightforward intuition about the construction process, deliberately omitting certain significant details that will be formally defined in Definition 6.6. The augmented game explicitly represents players' choices regarding whether to adhere to mediators' recommendations or to deviate from them. Consequently, the augmented games will have different structures depending on which solution concept is desired—we will define one augmented game  $\Gamma^c$  for each of our target solution concepts  $c$ . Figure 14 summarizes the main connections between the computation of an optimal correlated concept  $c$  (for instance, EFCE) in the original game  $\Gamma$ , and the computation of a Stackelberg equilibrium in the mediator-augmented game  $\Gamma^c$  corresponding to  $c$ . Figure 15 depicts the augmented games derived from the example of Figure 12 for the three solution concepts of interest.

In all three augmented games, the mediator has *imperfect recall*. This is crucial to correctly capture the correlated solution concepts. The imperfect recall is necessary for the one-to-one correspondence between *mixed strategies for the mediator* in the augmented game, and *correlated profiles of the players* in the original game. Intuitively, this is because the mediator's decisions in the augmented game correspond to *recommendations* in the original game, and therefore the mediator must pick one and only one recommendation in each information set. Thus, the mediator must have one infoset in the augmented game corresponding to each infoset in the original game. If the mediator were to have perfect recall, it would have the ability to “break” information sets by sending recommendations to a player that depend on information not known to



**Figure 14:** Correspondence between notions in the mediator-augmented game, and notions in the original game.

that player. Therefore, there could be a strategy for the mediator that does not correspond to a strategy profile in the original game.

**NFCCE.** In the case of NFCCE (Figure 15, top), the augmented game has an initial phase in which  $\blacktriangle$  and  $\blacktriangledown$  decide whether to deviate or obey to the mediator. Only one player is allowed to deviate in the game. When player  $\blacktriangle$  (resp.,  $\blacktriangledown$ ) deviates, all subsequent infosets will belong to either  $\blacktriangle$  (resp.,  $\blacktriangledown$ ) or to the mediator. The mediator takes decisions on behalf of the obedient player. If both players are obedient (see the subtree with leaf nodes  $p, q, r, s, j, k, l, m, n, o$ ), then all decisions after the initial phase are taken by the mediator.

**EFCCCE.** In the case of EFCCCE (Figure 15, middle) we can reason as follows: starting from the root of the original game  $\Gamma$ , we replace each information set of player  $\blacktriangle$  or  $\blacktriangledown$  with three new infosets. The first one is a parent info set modelling the decision of the player to obey or to deviate at the original info set in  $\Gamma$ . The two children information sets encode the decision to be taken at the original info set of  $\Gamma$  being replaced. The new info set following from the decision of the player to obey (at the parent info set) belongs to the mediator, who takes the action on behalf of the player. The new info set following from the decision of the player to deviate (at the parent info set) belongs to the deviating player, and it allows them for choosing the desired deviation. As before, after one player deviated, all the subsequent information sets belong to that player or to the mediator.

**EFCE.** In the case of EFCE (Figure 15, bottom), each of the original infosets  $I$  of  $\Gamma$  is duplicated and preceded by an information set of the mediator explicitly encoding the recommendation being issued at  $I$ . After observing the recommendation, the player decides whether to deviate or not. We note that actions within the same mediator's information set can represent recommendations as well as actions taken on behalf of the player. This is contingent upon whether the other player previously made a deviation or not. The information available to the mediator when recommending actions or acting on behalf of a player remains identical to what the player would have had in the original game.

Following this intuition, given a game  $\Gamma$ , we define the *augmented game*  $\Gamma^c$  corresponding to solution concept  $c$  as follows.

**Definition 6.6.** Given an extensive-form game  $\Gamma$ , a solution concept  $c$ , and an objective function  $g : \mathcal{Z} \rightarrow \mathbb{R}$ , the *augmented game*  $\Gamma^c$  is defined as follows.

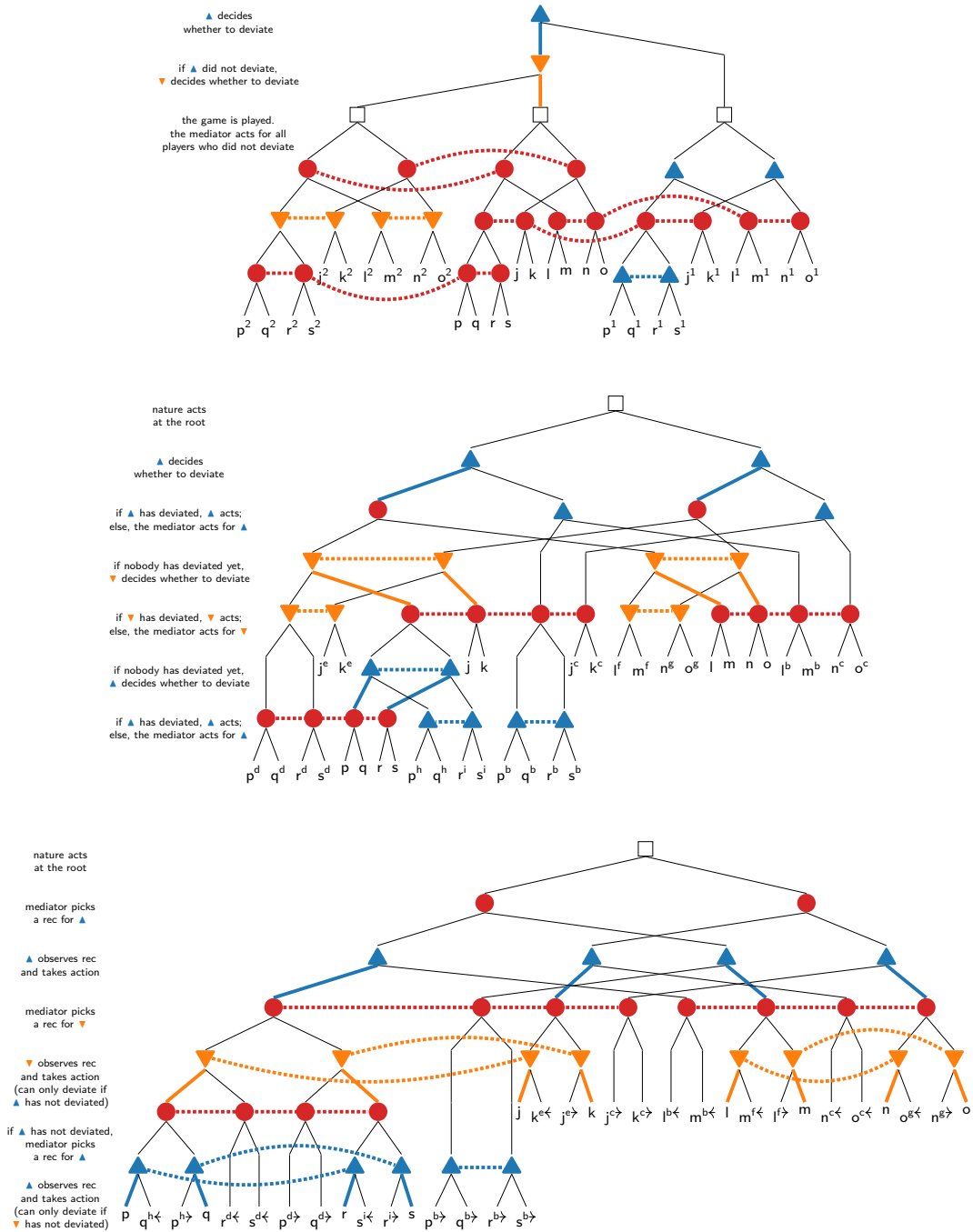
- *Players.*  $\Gamma^c$  has  $n + 1$  players: the  $n$  players in  $\Gamma$ , and a *mediator*.
- *Histories.* Unless otherwise stated, histories in  $\Gamma^c$  are identified with tuples  $(h, a, \tau)$ , where:
  - $h$  is a history in  $\Gamma$ ,
  - $a$  is either nothing ( $\perp$ ), a special symbol  $*$ , or an action  $a \in \mathcal{A}(h)$ , and
  - $\tau$  is either nothing ( $\perp$ ) or a trigger.

Intuitively, the three components of the history represent the following.

- $h$  is the true history of the game, representing the actions that have been taken by the players.
- $a$  is the recommendation from the mediator at the current infoset.  $\perp$  means that the mediator has yet to make a recommendation.  $*$  means that the mediator has not given a recommendation.
- $\tau$  represents the trigger, if any, that has been activated. Since we need only consider deviations of one player at a time, there can be at most one active trigger— $\perp$  means there is no active trigger.
- *Root node and NFCCE preplay phase.* If  $c \neq \text{NFCCE}$ , the root node of  $\Gamma$  is  $(\emptyset, \perp, \perp)$ . If  $c = \text{NFCCE}$ , there is a pre-play phase in which each of the  $n$  players, in order, chooses whether to deviate or not. Only one player may deviate: if player  $i$  deviates, then players  $j > i$  are forced to not deviate. Hence this pre-play phase is a tree with  $n + 1$  layers and  $n + 1$  leaves. The leaf in which player  $i$  deviates is  $(\emptyset, \perp, \emptyset_i)$ , and the leaf in which no player deviates is  $(\emptyset, \perp, \perp)$ .
- *Terminal nodes.* If  $z$  is terminal in  $\Gamma$ , then  $(z, \perp, \tau)$  is terminal in  $\Gamma^c$  for any  $\tau$ . At terminal node  $z$ , each player  $i$  receives utility  $u_i(z)$ , and the mediator receives utility  $g(z)$ .
- *Chance nodes.* If  $h$  is a chance node in  $\Gamma$ , then for any  $\tau$ ,  $(h, \perp, \tau)$  is also a chance node in  $\Gamma^c$  with the same chance probabilities. If  $c = \text{NFCCE}$ , the next node is  $(ha, \perp, \tau)$ . If  $c \neq \text{NFCCE}$ , the next node is a dummy node at which the sole action leads to  $(ha, \perp, \tau)$ .<sup>43</sup>
- *Player nodes, part 1.* Let  $(h, \perp, \tau)$  be a history of  $\Gamma^c$  such that  $h \in \mathcal{H}_i$  and  $i \neq 0$ . The structure of the game depends on  $c$ :
  - $c = \text{NFCCE}$ : If  $\tau = \emptyset_i$ , then player  $i$  observes the infoset  $I \ni h$  and picks an action  $a \in \mathcal{A}(h)$ . Otherwise, the mediator acts by picking an action  $a \in \mathcal{A}(h)$ . The next node is  $(ha, \perp, \tau)$ .
  - $c = \text{EFCCE}$ : If  $\tau \neq \perp$ , then  $(h, \perp, \tau)$  is a chance node with only one action, leading to node  $(h, *, \tau)$ . If  $\tau = \perp$ , then player  $i$  observes the infoset  $I \ni h$  decides whether or not to deviate. If player  $i$  deviates, then the next node is  $(h, *, I)$ . Otherwise, the next node is  $(h, *, \perp)$ .
  - $c = \text{EFCE}$ : If  $\tau$  is a trigger of player  $i$ , then  $(h, \perp, \tau)$  is a chance node with only one action, leading to  $(h, *, \tau)$ . Otherwise, the mediator selects an action  $a \in \mathcal{A}(h)$ , and the next node is  $(h, a, \tau)$ .
- *Player nodes, part 2.* Let  $(h, a, \tau)$  be a history of  $\Gamma^c$  such that  $h \in \mathcal{H}_i$  and  $i \neq 0$ , and  $a \neq \perp$ . The structure of the game again depends on  $c$ :
  - $c = \text{NFCCE}$ : This is impossible: by construction  $a = \perp$  always.
  - $c = \text{EFCCE}$ : if  $\tau$  is a trigger of player  $i$ , then player  $i$  selects an action  $a \in \mathcal{A}(h)$ . Otherwise, the mediator selects an action  $a \in \mathcal{A}(h)$ . In either case the next node is  $(ha, \perp, \tau)$ .
  - $c = \text{EFCE}$ : if  $\tau \neq \perp$  is a trigger not belonging to player  $i$ , then  $(h, a, \tau)$  is a chance node with a single action, leading to  $(ha, \perp, \tau)$ . Otherwise, player  $i$  observes the infoset  $I \ni h$  and action recommendation  $a$ , and selects an action  $a' \in \mathcal{A}(h)$ . If  $a' = a$  or  $a = *$ , then the next node is  $(ha, \perp, \tau)$ ; otherwise, the next node is  $(ha', \perp, Ia)$ .
- *Information.* Players  $i$  other than the mediator have perfect recall, and their observations are specified in the above game description. The mediator does *not* have perfect recall: two histories  $(h_1, \cdot, \cdot)$  and  $(h_2, \cdot, \cdot)$  belong to the same infoset in  $\Gamma^c$  if and only if  $h_1$  and  $h_2$  belong to the same infoset in  $\Gamma$ .

For concreteness, in Figure 15 we show the augmented games derived from the example game in Figure 12 for all three solution concepts. For notational shorthand, we will use  $h^\tau$  to refer to the node in  $\Gamma^c$  corresponding to the mediator making a recommendation with history  $h$  and trigger  $\tau$ —that is,  $h^\tau = (h, \perp, \tau)$  when  $c = \text{NFCCE}$  or  $\text{EFCE}$ , and  $h^\tau = (h, *, \tau)$  when  $c = \text{EFCCE}$ .

<sup>43</sup>The sole purpose of this “dummy layer” is to synchronize the timing between  $\Gamma^c$  and  $\Gamma$ .



**Figure 15:** Augmented games  $\Gamma^c$  for NFCCE (top), EFCCE (center), and EFCE (bottom), where  $\Gamma$  is the example game in Figure 12. The obedient strategies  $\mathbf{o}_1, \mathbf{o}_2$  are given by the thick colored lines below  $\blacktriangle$  and  $\blacktriangledown$ 's decision points. Red circles denote decision points of the mediator. Augmented histories are labeled as  $h^\tau$ , where  $h$  is the true node and  $\tau$  is the trigger. If no superscript is present, there was no trigger. For cleanliness,  $\tau$  is abbreviated in all three diagrams. For NFCCE,  $\tau$  is the player  $i$  who deviated—for example,  $p^2$  means terminal node  $p$  was reached, but  $\blacktriangledown$  deviated. For EFCCE,  $\tau$  is the node at which the player deviated—for example,  $p^d$  means terminal node  $p$  was reached, but  $\blacktriangledown$  deviated at node  $d$ . For EFCE,  $\tau$  is the node at which the player deviated, followed by the recommendation ( $\leftarrow$  or  $\rightarrow$ ) given to the player at that node—for example,  $q^{h\leftarrow}$  means terminal node  $q$  was reached but  $\blacktriangle$  deviated after being recommended to play  $\leftarrow$  at  $h$ .

### 6.4.1 Optimal Correlation via the Augmented Game

We now discuss how to use the augmented game  $\Gamma^c$  to compute optimal correlated equilibria in  $\Gamma$ . We first make a few critical observations:

First, the mediator has exactly one information set corresponding to each information set of the original game  $\Gamma$ . Therefore, *pure strategies* of the mediator correspond to *pure profiles* in  $\Gamma$ , and *mixed strategies* of the mediator correspond to *correlated profiles* in  $\Gamma$ . We will therefore abuse notation and also use  $\xi$  to refer to mixed strategies for the mediator in  $\Gamma$ . Critically, the sequence form of  $\xi$  in each augmented game will have enough information about the correlated distribution to define the incentive constraints of the players. Second, each player has a unique *obedient strategy*  $\mathbf{o}_i$ , defined by always obeying recommendations (for EFCE) and never choosing to deviate (or NFCCE and EFCCE). Finally, the size of the  $\Gamma^c$  is polynomial in the size of  $\Gamma$ .

As a notational convention, where context is insufficient, we will generally use a superscript  $c$  to distinguish the augmented game from the original game—for example,  $\mathcal{X}_i^c$  will denote the strategy set of player  $i$  in  $\Gamma^c$ , *etc.*

Now let  $\xi$  be a mediator mixed strategy in  $\Gamma^c$ . Then  $\xi$  represents an equilibrium in  $\Gamma$  if and only if, in the profile  $(\xi, \mathbf{o}_1^c, \dots, \mathbf{o}_n^c)$ , each (non-mediator) player  $i$  is playing a best response. That is, solving the following program will give an optimal equilibrium:

$$\max_{\xi \in \Xi^c} g(\xi) \quad \text{s.t.} \quad \max_{\mathbf{x}_i^c \in \text{co } \mathcal{X}_i^c} u_i(\xi, \mathbf{x}_i^c, \mathbf{o}_{-i}^c) \leq u_i(\xi, \mathbf{o}_i^c, \mathbf{o}_{-i}^c) \quad \forall i \in [n]$$

where  $\Xi^c$  is the mediator's sequence-form mixed strategy set in  $\Gamma^c$ , and  $\text{co } \mathcal{X}_i^c$  is player  $i$ 's mixed strategy set in  $\Gamma^c$ . Now, by representing the mixed strategy of each player (including the mediator) in sequence form, the utility functions are linear in each strategy. Therefore, the above program can be rewritten as

$$\max_{\xi \in \Xi^c} \mathbf{g}^\top \xi \quad \text{s.t.} \quad \max_{\mathbf{x}_i^c \in \text{co } \mathcal{X}_i^c} \xi^\top \mathbf{A}_i \mathbf{x}_i^c \leq \mathbf{b}_i^\top \xi \quad \forall i \in [n] \quad (6)$$

for vectors and matrices  $\mathbf{g}$ ,  $\mathbf{A}_i$ , and  $\mathbf{b}_i$ . Now, the inner maximization

$$\max_{\mathbf{x}_i^c \in \text{co } \mathcal{X}_i^c} \xi^\top \mathbf{A}_i \mathbf{x}_i^c \quad (7)$$

is itself an LP where  $\xi$  is a constant. Moreover, since each player  $i$  has perfect recall, the sequence-form strategy sets  $\text{co } \mathcal{X}_i^c$  can be represented as polytopes  $\text{co } \mathcal{X}_i^c = \{\mathbf{x}_i^c \geq \mathbf{0} : \mathbf{F}_i^c \mathbf{x}_i^c = \mathbf{f}_i^c\}$  for matrix and vector  $\mathbf{F}_i^c, \mathbf{f}_i^c$  of size linear in the size of  $\Gamma^c$ . We therefore can formally take a dual of (7), resulting in the LP

$$\min_{\mathbf{v}_i} (\mathbf{f}_i^c)^\top \mathbf{v}_i \quad \text{s.t.} \quad \mathbf{A}_i^\top \xi \leq (\mathbf{F}_i^c)^\top \mathbf{v}_i. \quad (8)$$

By strong duality of linear programs (which holds in this case because (7) is always feasible), the programs (7) and (8) have the same value. Therefore, (6) is equivalent to the linear program

$$\begin{cases} \max_{\xi, \mathbf{v}_i: i \in [n]} \mathbf{g}^\top \xi \\ \text{s.t.} \quad \textcircled{1} \mathbf{A}_i^\top \xi \leq (\mathbf{F}_i^c)^\top \mathbf{v}_i \quad \forall i \in [n] \\ \quad \quad \textcircled{2} (\mathbf{f}_i^c)^\top \mathbf{v}_i \leq \mathbf{b}_i^\top \xi \quad \forall i \in [n] \\ \quad \quad \textcircled{*} \xi \in \Xi^c \end{cases} \quad (9)$$

This program has size linear in the size of  $\Gamma^c$  and the description of the polytope  $\Xi^c$ . Unfortunately, in general, since the mediator has imperfect recall, there is no efficient way of representing  $\Xi^c$ , that is, there is no polynomial system of linear constraints describing  $\Xi^c$ . Indeed computing optimal equilibria for all three notions  $c$  is NP-hard (von Stengel and Forges, 2008).

Although the *pure strategy sets* for the mediator are essentially the same in all three augmented games, the *sequence-form strategy sets*  $\Xi^c$  are substantially different. The differences arise due to more deviations being possible for some notions than for others. Consider for example the game  $\Gamma$  depicted in Figure 12. In the



augmented game  $\Gamma^{\text{EFCE}}$  (Figure 15, bottom), there is a terminal node  $\mathbf{p}^{\mathbf{b}\triangleright}$  whose player reach probability  $\xi[\mathbf{p}^{\mathbf{b}\triangleright}]$  is the probability that the mediator recommends  $\blacktriangle$  to play right at  $\mathbf{b}$  and  $\blacktriangledown$  to play left at infoset  $\mathbf{de}$ . There is no node in  $\Gamma^{\text{NFCCE}}$  whose player reach probability represents the same thing. It should therefore remain intuitively plausible that  $\Xi^{\text{EFCE}}$  should be more difficult to represent than  $\Xi^{\text{NFCCE}}$ . In the next section, we will discover that this is precisely the case.

## 6.4.2 Comparison to Relevant Sequence-Based Construction of $\Xi$

Our construction via the mediator-augmented game uses a vector  $\xi \in \Xi^c$  to represent a correlated profile. It is instructive to compare this representation to other representations of correlated profiles, in particular, the *correlation plan* defined and used by von Stengel and Forges (2008). In this section, we will review the notion of correlation plan defined by that paper, and compare it to our construction.

**Definition 6.7.** A sequence tuple  $(I_1 a_1, \dots, I_n a_n) \in \Sigma_1 \times \dots \times \Sigma_n$  is *relevant* if there is a history  $h$  in  $\Gamma$  such that either  $\sigma_i(h) = I_i a_i$  for every player  $i$ , or there is a player  $j$ —the *deviator*—such that  $\sigma_i(h) = I_i a_i$  for all  $i \neq j$  and  $I_j \preceq h$ .

This definition was first proposed by von Stengel and Forges (2008) in the two-player case; here, we generalize it to arbitrarily many players. Intuitively, the relevant tuples are those that appear in the linear program defining *any* of the three notions.

**Definition 6.8** (von Stengel and Forges, 2008). For a correlated profile  $\mu \in \Delta(X_1 \times \dots \times X_n)$ , the *correlation plan* is the vector  $\xi \in \mathbb{R}^\Sigma$  defined by  $\xi[\sigma_1, \dots, \sigma_n] = \mathbb{E}_{\mathbf{x} \sim \mu} \prod_{i \in [n]} \mathbf{x}_i[\sigma_i]$ . We denote by  $\Xi$  the set of all correlation plans.

von Stengel and Forges (2008) go on to show that correlation plans are a sufficient representation for computing (optimal) EFCE, in the sense that, if one could efficiently represent the set of all correlation plans, then one can compute optimal EFCE efficiently. Farina et al. (Farina et al., 2019b, 2020) generalizes this observation to NFCCE and EFCCE as well. Our linear program (9) achieves the same claim: if  $\Xi^c$  is efficiently representable then optimal equilibria in notion  $c$  can be computed efficiently. One may wonder, therefore, about the relationship between the two.

It turns out that each of our  $\Xi^c$  polytopes is in some sense merely a sub-vector of  $\Xi$  with the indices renamed. That is, there is a natural injection from sequences of the mediator in  $\Gamma^c$  to relevant tuples  $(\sigma_1, \dots, \sigma_n) \in \Sigma$ . A mediator sequence in  $\Gamma^c$  corresponds to some history  $h^\tau$ . If  $\tau = \perp$  then  $h^\tau$  corresponds to  $(\sigma_1(h), \dots, \sigma_n(h))$ , that is,  $\xi[h^\tau] = \xi[\sigma_1(h), \dots, \sigma_n(h)]$ ; if  $\tau$  is a nonempty trigger (say, P1 WLOG), then  $h^\tau$  corresponds to  $(\sigma_1(\tau), \sigma_2(h), \dots, \sigma_n(h))$ , where  $\sigma_1(\tau)$  is the last sequence of player  $i$  before  $\tau$ . By construction of  $\Gamma^c$ , this must be a relevant tuple.

In some sense,  $\Xi^c$  is therefore a *refined* notion of correlation plan that is specific to the equilibrium concept  $c$ , only requiring the sequence tuples that are relevant for that concept. In the next section, we will show that, in fact, the differences between the various  $\Xi^c$ s result in separations in the complexity of representing each polytope, and therefore separations in the complexity of computing optimal equilibria.

The key barrier to computing optimal equilibria, in a sense, is that *the mediator in the augmented game has imperfect recall*. In the next two sections, we will describe two methods of overcoming this imperfect recall and thus of arriving at algorithms for computing optimal equilibria. The first (Section 6.5) applies the recent construction of Zhang et al. (2023b), which is a general method of representing the sequence form of an imperfect-recall player in a timeable game. The second (deferred to full paper (Zhang et al., 2024c)) is a variant of *column generation* which is most powerful in two-player games, in which one (and only one) player is allowed to play a *mixed* strategy, thereby allowing a much greater strategy set to be available for any given support.

## 6.5 Representing Imperfect-Recall Decision Spaces

Zhang et al. (2023b) recently developed a method for representing the sequence-form strategy spaces for imperfect-recall players (equivalently, teams of players who cannot communicate) in timeable games. Since the augmented games  $\Gamma^c$  are timeable, we directly apply their main result to our problem.

**Definition 6.9.** In a timeable extensive-form game  $\Gamma'$ , the *connectivity graph*  $G_S$  of a subset of players  $S \subseteq [n]$  is the graph whose nodes are histories of  $\Gamma'$ , and where there is an edge  $(h, h')$  if  $h$  and  $h'$  are in the same level of the tree, and they are *connected*, i.e., there is an info set  $I \in \mathcal{I}_i$ , where  $i \in S$ , with  $h, h' \preceq I$ .

**Definition 6.10.** A set of nodes  $B \subseteq \mathcal{H}$  is a *belief* for player  $i$  if

1.  $B$  contains at least one decision point for player  $i$ , that is,  $B \cap \mathcal{H}_i \neq \emptyset$
2. there exists a pure strategy  $\mathbf{x}_i$  for player  $i$  such that  $B$  is a connected component of  $G_i[\{h \in \mathcal{H} : x_i[h] = 1\}]$  where  $G_i[\cdot]$  denotes an induced connected component of  $G_i$ .

We will use  $\mathcal{B}_i$  to denote the set of beliefs of player  $i$ .

Intuitively, beliefs represent sets of nodes that an imperfect-recall player *will always be able to distinguish in the future*: that is, if  $B$  is a belief corresponding to pure strategy  $\mathbf{x}_i$ , then, upon reaching the belief  $B$ , player  $i$  knows that it has reached belief  $B$ , and player  $i$  knows that it will never forget having reached  $B$ . Recall from Section 3.5 the team belief DAG:

**Theorem 6.11** (Team Belief DAG). *There exists a representation of player  $i$ 's decision space as a polytope whose constraint matrix has  $O^*(R_i)$  entries, where*

$$R_i := \sum_{B \in \mathcal{B}_i} \prod_{\substack{I \in \mathcal{I}_i: \\ I \cap B \neq \emptyset}} |\mathcal{A}(I)| \quad (10)$$

The representation uses a DAG to model the decision problem faced by player  $i$ , and then bounds the number of nodes in the DAG. For intuition, when  $\Gamma'$  has perfect recall, one can check that beliefs are always disjoint and every info set  $I \in \mathcal{I}_i$  is a belief, so the above expression is linear in the size of the game—indeed, in that case, the representation reduces to the sequence-form polytope.

We use the above result to construct a representation of the mediator's decision space,  $\Xi^c$ , in the augmented game  $\Gamma' := \Gamma^c$ . We call the representation of  $\Xi^c$  using Theorem 6.11 the *correlation DAG* for notion  $c$ . Theorem 6.11 immediately gives an algorithm for solving the program (9). This algorithm is given in Algorithm **CorrelationDAG**.

---

### Algorithm CorrelationDAG: Optimal Correlated Equilibria via Correlation DAG

---

- 1: **input:** extensive-form game  $\Gamma$ , desired solution concept  $c$ , objective  $g : \mathcal{Z} \rightarrow \mathbb{R}$
  - 2: construct the augmented game  $\Gamma^c$
  - 3: compute a polytope representation of the mediator's strategy space,  $\Xi^c$ , using Theorem 6.11
  - 4: solve the LP (9)
  - 5: **return**  $\xi$
-

### 6.5.1 Analyzing the Size of the Representation

To analyze the complexity of Algorithm **CorrelationDAG**, it suffices to bound the quantity in (10). Notationally, we will use  $R_M^c$  to denote the quantity  $R_M$  in (10) in the augmented game  $\Gamma^c$ . We first introduce some useful definitions.

**Definition 6.12.** A *public state* is a connected component of  $G = G_{[n]}$ .

**Definition 6.13.** Given a node  $h$  and a player  $i$ , the *last info set*  $I_i(h)$  is the lowest (*i.e.*, most recent) info set reached by player  $i$  on the path to  $h$ .

**Definition 6.14.** The *information complexity*  $k$  of an extensive-form game is the greatest number of unique last info sets in any public state. In symbols,  $k = \max_{P \in \mathcal{P}} |\{I_i(h) : h \in P, i \in [n]\}|$ .

Notice that it is possible for  $k$  to be much smaller than  $n|P|$ , because the set of last info sets may contain duplicates. For example, in normal-form games (converted to extensive form in the canonical manner), we have  $k = n$  since each terminal node is a public state and each player has only one info set. As an example, the information complexity of the game in Figure 12 is 3: the public state **de** has three last info sets, namely **b**, **c**, and **de** itself.

Zhang et al. (2023b) use the definition of information complexity to bound the representation size of Theorem 6.11. In particular, they show that if the decision problem for the imperfect-recall player  $i$  can be decomposed into  $n$  perfect-recall players such that the information complexity is  $k$ , then  $R_i \leq O^*((b+1)^k)$ , where  $b$  is the branching factor of the game. In this section, we show similar bounds in our setting. Note that  $b$  and  $k$  here are the branching factor and information complexity of the original game  $\Gamma$ , not of  $\Gamma^c$ —therefore, we cannot directly apply the bound  $R_i \leq O^*((b+1)^k)$ . Indeed, the mediator in  $\Gamma^c$  can have much higher information complexity than  $\Gamma$ . Thus, we need to be more careful in our analysis.

**Theorem 6.15.** *Let  $k$  be the information complexity of a timeable game  $\Gamma$ ,  $b$  be its branching factor, and  $d$  be its depth. Then  $R_M^{\text{NFCCE}} \leq O^*((b+1)^k)$ ,  $R_M^{\text{EFCCE}} \leq O^*((b+d-1)^k)$ , and  $R_M^{\text{EFCCE}} \leq O^*((bd)^k)$ .*

As an example, consider an extensive-form game of the following form. Chance first samples and privately reveals types  $t_i \in [T]$  to each player  $i$ . Thereafter, there is no further privacy: all actions by the players and chance after the root are public. By definition, we see that this game is a public-action game, and we have  $k = nT$  because each sequence of post-root actions induces a public state with  $T$  private states for each of the  $n$  players. Thus, Theorem 6.15 gives an algorithm for computing optimal EFCEs that runs in time  $\text{poly}(|\mathcal{H}|, (bd)^{nT})$ ; in particular, if  $n = T = O(1)$  then the algorithm runs in polynomial time. To our knowledge, we are the first to give a polynomial-time algorithm for this setting, even when  $n = T = 2$ .

We now show two settings in which we can improve our bounds from Theorem 6.15. They both depend on certain information being *public*.

### 6.5.2 Public Player Actions

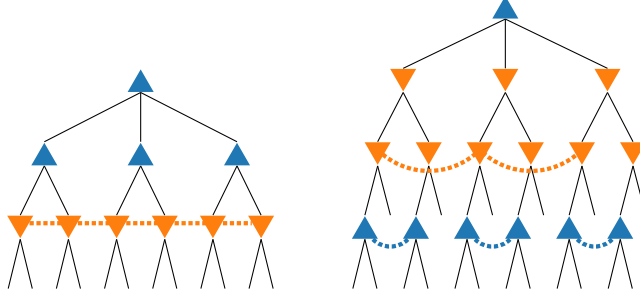
First, we discuss the setting in which *player* actions are public.

**Definition 6.16.** A game has *public player actions* if, for all public states  $P \in \mathcal{P}$  containing at least one non-chance node, for all actions  $a \in \bigcup_{h \in P} \mathcal{A}(h)$ , the set  $\{ha : h \in P, a \in \mathcal{A}(h)\}$  is a union of public states.

Poker, for example, has this structure: the root public state contains only a chance node, and every action thereafter is fully public. In this setting, we can remove the dependencies on  $b$  for NFCCE and EFCCE:

**Theorem 6.17.** *In games with public player actions,  $R_M^{\text{NFCCE}} = O^*(3^k)$  and  $R_M^{\text{EFCCE}} = O^*(d^k)$ .*

Intuitively, the proof works by constructing a new game that reduces the branching factor of the original game to 2 while keeping all other relevant structure intact. The fact that the players' actions are public



**Figure 16:** Two examples of two-player extensive-form game trees with no chance moves and large information complexity  $k$ . In both examples,  $k$  can be increased arbitrarily by increasing the branching factor of the root node. The left example would be easily repairable with a tighter definition of information complexity (that takes into account the fact that only one of the infosets in the second layer is reachable in any pure strategy profile), but the right example is not so easily repairable, and examples such as these are the reason that the proof of Theorem 6.19 is more involved than one may initially expect.

ensures that this transformation does not increase  $k$ . We defer the full proof to the full paper (Zhang et al., 2024c).

Once again, the bound for NFCCE matches that of Zhang et al. (2023b) in team games, up to polynomial factors. The bound on  $R_M^{\text{EFCE}}$  cannot be improved in this fashion, for two reasons. First, the  $(bd)^k$  term in that analysis comes from counting the number of triggers at a given node, which has not changed. Second, as above, the proof of Theorem 6.17 modifies the original game tree to have lower branching factor. This is an invalid transformation for EFCE, because some EFCE triggers present in the original game would not be expressible in the new game.

## 6.6 Two-Player Games with Public Chance

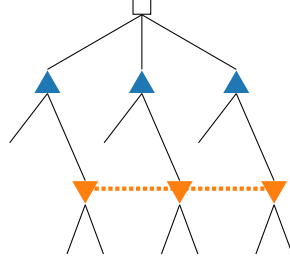
We now discuss the case where *chance* actions are public. Since it is already NP-hard to compute optimal equilibria in three-player games with no chance nodes (von Stengel and Forges, 2008), we restrict our attention to *two-player* games. Farina and Sandholm (2020) showed, via a different construction, that in games with public chance,  $\Xi$  has a polynomial-sized representation and therefore optimal NFCCEs, EFCCEs, and EFCEs can be computed in polynomial time. In this section, we show that our correlation DAG matches this bound.

**Definition 6.18.** A game has *public chance actions* if, for every two nodes  $h, h'$  in the same public state, the lowest common ancestor  $h \wedge h'$  is not a chance node.

We will assume for the rest of this section that levels in  $\Gamma$  uniquely specify whose move it is—that is, for every level of the game tree, there exists a player  $i$  (possibly nature) such that every node in the level is a decision node of player  $i$ . Since we have already assumed timeability, this additional assumption is without loss of generality by adding dummy nodes (Carminati et al., 2022). Most practical games, including the games we use in our experiments, already satisfy this assumption without further modification.

**Theorem 6.19.** *In two-player timeable games with public chance actions, we have  $R_M^c = \text{poly}(|\mathcal{H}|)$  for all three notions  $c$ .*

Initially, one may ask whether it is possible to prove this result by directly applying Theorem 6.15. In particular, if it were the case that all two-player games of public chance had constant information complexity, Theorem 6.19 would follow immediately. Unfortunately, this is not the case: in Figure 16, we exhibit two families of two-player extensive-form games with *no* chance actions and information complexity that is linear in the size of the game.



**Figure 17:** An example of a timeable triangle-free game in which our construction will be exponentially-sized (in the branching factor of the root node). In this game, the algorithm of Farina and Sandholm (2020) works by essentially “re-ordering” the game tree so that  $\blacktriangledown$ ’s decision point is moved to the root, at which point the chance decision can be treated as public, thereby removing the exponentiality.

### 6.6.1 Discussion: Relationship to Triangle-Freeness

Theorem 6.19 implies that Algorithm **CorrelationDAG** runs in polynomial time in two-player games of public chance. As we mentioned, we are not the first to exhibit a polynomial-time algorithm in this setting; Farina and Sandholm (2020) has exhibited one using a different technique, namely by showing that the *von Stengel–Forges* (vSF) polytope (von Stengel and Forges, 2008) is tight. It is instructive to compare the two approaches. The approach of Farina and Sandholm (2020) carries many similarities to our approach for this special case—in particular, their approach also works by effectively constructing a DAG representation of  $\Xi^{\text{EFCE}}$ . However, while their approach dynamically chooses which information set to expand next on the fly, our approach uses the fixed ordering provided by the timeable game to decide which information set is “next”. When the game is timeable, our approaches give essentially the same representation: indeed, the proof in the previous section shows that there is a decision point of the mediator in  $\Gamma^c$  for every relevant pair  $(I_1, \sigma_2)$  or  $(\sigma_1, I_2)$ , which are precisely the branching points in the representation of Farina and Sandholm (2020).

Unlike their approach, our correlation DAG algorithm provides an FPT guarantee on any game. However, it is limited to timeable games, whereas theirs generalizes beyond timeable games to a family they coin *triangle-free games*. Here, for the sake of completeness, we include a definition of triangle-freeness.

**Definition 6.20.** In a two-player game, two information sets  $I_1 \in \mathcal{I}_1$  and  $I_2 \in \mathcal{I}_2$  are *connected*, denoted  $I_1 \bowtie I_2$ , if there exists a node  $h$  with  $h \succeq I_1$  and  $h \succeq I_2$ . A *triangle* is a collection of four infosets  $I_1, I'_1 \in \mathcal{I}_1$  and  $I_2, I'_2 \in \mathcal{I}_2$  such that  $I_1 \bowtie J_1$ ,  $I_2 \bowtie J_2$ , and  $I_1 \bowtie J_2$ .

Intuitively, triangle-freeness is useful because it guarantees the existence of some “branching order” that can be used to fill in the polytope  $\Xi^{\text{EFCE}}$ . We refer the reader to the paper of Farina and Sandholm (2020) for more details. It is not difficult to construct triangle-free games in which our construction would be exponentially-sized; see Figure 17. We leave to future research the question of whether it is possible to extend our algorithm so that it is also runs in polynomial time in all triangle-free games, achieving the best of both worlds.

### 6.6.2 Fixed-Parameter Hardness of Representing $\Xi^{\text{EFCCE}}$ and $\Xi^{\text{EFCE}}$

A natural question is whether it is possible to achieve the same bound for EFCCE and EFCE as achieved for NFCCE and team games—namely, a construction whose exponential term depends only on  $b$  and  $k$ . It turns out that our construction does *not* accomplish this, and in fact, *no* representation of  $\Xi^c$  for  $c = \text{EFCCE}$  or  $c = \text{EFCE}$  can have size  $O^*(f(k))$  for any function  $f$  under standard complexity assumptions even when  $b = 2$ . To do this, we first review some fundamental notions of *parameterized complexity*.

**Definition 6.21.** A *fixed-parameter tractable* (FPT) algorithm for a problem is an algorithm that takes as input an instance  $x$  and a *parameter*  $k \in \mathbb{N}$ , and runs in time  $f(k)\text{poly}(|x|)$ , where  $|x|$  is the bit length of  $x$  and  $f : \mathbb{N} \rightarrow \mathbb{N}$  is an *arbitrary function*.

The  $k$ -CLIQUE problem<sup>44</sup> is widely conjectured to not admit an FPT algorithm parameterized by the clique size  $k$ . In the literature on parameterized complexity, this conjecture is known as  $FPT \neq W[1]$ , and is implied by the exponential time hypothesis (Chen et al., 2005). We now show that this conjecture implies lower bounds on the complexity of representing the polytopes  $\Xi^{\text{EFCCE}}$  and  $\Xi^{\text{EFCE}}$ .

**Theorem 6.22.** *Assuming  $FPT \neq W[1]$ , there is no FPT algorithm for linear optimization over  $\Xi^{\text{EFCCE}}$  or  $\Xi^{\text{EFCE}}$  parameterized by information complexity, even in two-player games with constant branching factor.*

Technically speaking, this result does not establish parameterized hardness of computing optimal EFCCes or EFCEs, as there could hypothetically be a method for doing so that exploits the special nature of the (9). Indeed, the proof of Theorem 6.22 exploits the fact that the objective coefficient  $g[h^\tau]$  may depend on  $\tau$  as well as  $h$ , which is not the case for the LP (9). However, we know of no technique for optimal equilibria that would not also imply the ability to optimize over  $\Xi^c$ . Therefore, Theorem 6.22 is a lower bound that applies to all known techniques for computing optimal EFCCes and EFCEs.

## 7 Computing Optimal Equilibria and Mechanisms via Learning in Zero-Sum Games

### 7.1 Introduction

In this paper, we introduce a new paradigm of learning in games for *computing* optimal equilibria. It applies to extensive-form settings with any number of players, including information design, and solution concepts such as correlated, communication, and certification equilibria. Further, our framework is general enough to also capture optimal mechanism design and optimal incentive design problems in sequential settings.

**Summary of Our Results.** A key insight that underpins our results is that computing *optimal* equilibria in multi-player extensive-form games can be cast via a Lagrangian relaxation as a two-player zero-sum extensive-form game. This unlocks a rich technology, both theoretical and experimental, developed for computing minimax equilibria for the more challenging—and much less understood—problem of computing optimal equilibria. In particular, building on the framework of Zhang and Sandholm (2022a), our reduction lends itself to mechanism design and information design, as well as an entire hierarchy of equilibrium concepts, including *normal-form coarse correlated equilibria (NFCCE)* (Moulin and Vial, 1978), *extensive-form coarse correlated equilibria (EFCCE)* (Farina et al., 2020), *extensive-form correlated equilibria (EFCE)* (von Stengel and Forges, 2008), *communication equilibria (COMM)* (Forges, 1986; Myerson, 1986), and *certification equilibria (CERT)* (Forges and Koessler, 2005). In fact, for communication and certification equilibria, our framework leads to the first learning-based algorithms for computing them, addressing a question left open by Zhang and Sandholm (2022a) (*cf.* (Fujii, 2023)).

We thus focus on computing an optimal equilibrium by employing regret minimization techniques in order to solve the induced bilinear saddle-point problem. Such considerations are motivated in part by the remarkable success of no-regret algorithms for computing minimax equilibria in large two-player zero-sum games (*e.g.*, see (Bowling et al., 2015; Brown and Sandholm, 2018)), which we endeavor to transfer to the problem of computing optimal equilibria in multi-player games.

In this context, we show that employing standard regret minimizers, such as online mirror descent (Shalev-Shwartz, 2012) or counterfactual regret minimization (Zinkevich et al., 2007), leads to a rate of convergence of  $T^{-1/4}$  to optimal equilibria by appropriately tuning the magnitude of the Lagrange multipliers (Corollary 7.4). We also leverage the technique of *optimism*, pioneered by Chiang et al. (2012); Rakhlin and Sridharan (2013b) and Syrgkanis et al. (2015), to obtain an accelerated  $T^{-1/2}$  rate of convergence (Corollary 7.5). These are the first learning dynamics that (provably) converge to optimal equilibria. Our bilinear formulation also allows us

<sup>44</sup>The  $k$ -CLIQUE problem is to decide whether a given graph contains a clique of size at least  $k$ .

to obtain *last-iterate* convergence to optimal equilibria via optimistic gradient descent/ascent (Theorem 7.6), instead of the time-average guarantees traditionally derived within the no-regret framework. As such, we bypass known barriers in the traditional learning paradigm by incorporating an additional player, a *mediator*, into the learning process. Furthermore, we also study an alternative Lagrangian relaxation which, unlike our earlier approach, consists of solving a sequence of zero-sum games (*cf.* (Farina et al., 2019b)). While the latter approach is less natural, we find that it is preferable when used in conjunction with deep RL solvers since it obviates the need for solving games with large reward ranges—a byproduct of employing the natural Lagrangian relaxation.

**Experimental results.** We demonstrate the practical scalability of our approach for computing optimal equilibria and mechanisms. First, we obtain state-of-the-art performance in a suite of 23 different benchmark game instances for seven different equilibrium concepts. Our algorithm significantly outperforms existing LP-based methods, typically by more than one order of magnitude. We also use our algorithm to derive an optimal mechanism for a sequential auction design problem, and we demonstrate that our approach is naturally amenable to modern deep RL techniques.

## 7.2 Preliminaries

We adopt the general framework of *mediator-augmented games* of Zhang and Sandholm (2022a) to define our class of instances. Thus, for us, an extensive-form game has  $n$  players, a mediator  $M$ , and chance  $C$ . The *mixed* realization-form strategy set of the mediator is denoted  $\Xi$ .

**Revelation principle.** The *revelation principle* allows us, without loss of generality, to restrict our attention to equilibria where each player is playing some fixed pure strategy  $\mathbf{o}_i \in \mathcal{X}_i$ .

**Definition 7.1.** The game  $\Gamma$  satisfies the *revelation principle* if there exists a *direct* pure strategy profile  $\mathbf{o} = (\mathbf{o}_1, \dots, \mathbf{o}_n)$  for the players such that, for all strategy profiles  $(\boldsymbol{\mu}, \mathbf{x})$  for all players including the mediator, there exists a mediator strategy  $\boldsymbol{\mu}' \in \Xi$  and functions  $f_i : \mathcal{X}_i \rightarrow \mathcal{X}_i$  for each player  $i$  such that:

1.  $f_i(\mathbf{o}_i) = \mathbf{x}_i$ , and
2.  $u_j(\boldsymbol{\mu}', \mathbf{x}'_i, \mathbf{o}_{-i}) = u_j(\boldsymbol{\mu}, f_i(\mathbf{x}'_i), \mathbf{x}_{-i})$  for all  $\mathbf{x}'_i \in \mathcal{X}_i$ , and players  $j \in [n] \cup \{M\}$ .

The function  $f_i$  in the definition of the revelation principle can be seen as a *simulator* for Player  $i$ : it tells Player  $i$  that playing  $\mathbf{x}'_i$  if other players play  $(\boldsymbol{\mu}, \mathbf{o}_{-i})$  would be equivalent, in terms of all the payoffs to all agents (including the mediator), to playing  $f(\mathbf{x}'_i)$  if other agents play  $(\boldsymbol{\mu}, \mathbf{x}_{-i})$ . It follows immediately from the definition that if  $(\boldsymbol{\mu}, \mathbf{x})$  is an  $\epsilon$ -equilibrium, then so is  $(\boldsymbol{\mu}', \mathbf{o})$ —that is, every equilibrium is payoff-equivalent to a direct equilibrium.

The revelation principle applies and covers many cases of interest in economics and game theory. For example, in (single-stage or dynamic) mechanism design, the direct strategy  $\mathbf{o}_i$  of each player is to report all information truthfully, and the revelation principle guarantees that for all non-truthful mechanisms  $(\boldsymbol{\mu}, \mathbf{x})$  there exists a truthful mechanism  $(\boldsymbol{\mu}', \mathbf{o})$  with the same utilities for all players.<sup>45</sup> For correlated equilibrium, the direct strategy  $\mathbf{o}_i$  consists of obeying all (potentially randomized) recommendations that the mediator gives, and the revelation principle states that we can, without loss of generality, consider only correlated equilibria where the signals given to the players are what actions they should play. In both these cases (and indeed in general for the notions we consider in this paper), it is therefore trivial to specify the direct strategies  $\mathbf{o}$  without any computational overhead. Indeed, we will assume throughout the paper that the direct strategies  $\mathbf{o}$  are given. Further examples and discussion of this definition can be found in the appendix of the full paper (Zhang et al., 2023a).

Although the revelation principle is a very useful characterization of optimal equilibria, as long as we are given  $\mathbf{o}$ , all of the results in this paper actually apply regardless of whether the revelation principle is satisfied: when it fails, our algorithms will simply yield an *optimal direct equilibrium* which may not be an optimal equilibrium.

<sup>45</sup>In a mechanism design context, a strategy for the mediator  $\boldsymbol{\mu}$  induces a mechanism; here we slightly abuse terminology by referring to  $(\boldsymbol{\mu}, \mathbf{d})$  also as a mechanism.

Under the revelation principle, the problem of computing an optimal equilibrium can be expressed as follows:

$$\max_{\boldsymbol{\mu} \in \Xi} u_M(\boldsymbol{\mu}, \boldsymbol{o}) \quad \text{s.t.} \quad \max_{\mathbf{x}_i \in \text{co } \mathcal{X}_i} u_i(\boldsymbol{\mu}, \mathbf{x}_i, \boldsymbol{o}_{-i}) \leq u_i(\boldsymbol{\mu}, \boldsymbol{o}) \quad \forall i \in [n].$$

The objective  $u_M(\boldsymbol{\mu}, \boldsymbol{o})$  can be expressed as a linear expression  $\mathbf{c}^\top \boldsymbol{\mu}$ , and  $u_i(\boldsymbol{\mu}, \mathbf{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{\mu}, \boldsymbol{o})$  can be expressed as a bilinear expression  $\boldsymbol{\mu}^\top \mathbf{A}_i \mathbf{x}_i$ . Thus, the above program can be rewritten as

$$\max_{\boldsymbol{\mu} \in \Xi} \mathbf{c}^\top \boldsymbol{\mu} \quad \text{s.t.} \quad \max_{\mathbf{x}_i \in \text{co } \mathcal{X}_i} \boldsymbol{\mu}^\top \mathbf{A}_i \mathbf{x}_i \leq 0 \quad \forall i \in [n]. \quad (11)$$

Zhang and Sandholm (2022a) now proceed by taking the dual linear program of the inner maximization, which suffices to show that (11) can be solved using linear programming.<sup>46</sup>

Finally, although our main focus in this paper is on games with discrete action sets, it is worth pointing out that some of our results readily apply to continuous games as well using, for example, the discretization approach of Kroer and Sandholm (2015).

## 7.3 Lagrangian Relaxations and a Reduction to a Zero-Sum Game

Our approach in this paper relies on Lagrangian relaxations of the linear program (11). In particular, in this section we introduce two different Lagrangian relaxations. The first one (Section 7.3.1) reduces computing an optimal equilibrium to solving a *single* zero-sum game. We find that this approach performs exceptionally well in benchmark extensive-form games in the tabular regime, but it may struggle when used in conjunction with deep RL solvers since it increases significantly the range of the rewards. This shortcoming is addressed by our second method, introduced in Section 7.3.2, which instead solves a *sequence* of suitable zero-sum games.

### 7.3.1 “Direct” Lagrangian

Directly taking a Lagrangian relaxation of the LP (11) gives the following saddle-point problem:

$$\max_{\boldsymbol{\mu} \in \Xi} \min_{\substack{\lambda \in \mathbb{R}_{\geq 0}, \\ \mathbf{x}_i \in \text{co } \mathcal{X}_i: i \in [n]}} \mathbf{c}^\top \boldsymbol{\mu} - \lambda \sum_{i=1}^n \boldsymbol{\mu}^\top \mathbf{A}_i \mathbf{x}_i. \quad (\text{L1})$$

We first point out that the above saddle-point optimization problem admits a solution  $(\boldsymbol{\mu}^*, \mathbf{x}^*, \lambda^*)$ :

**Proposition 7.2.** *The problem (L1) admits a finite saddle-point solution  $(\boldsymbol{\mu}^*, \mathbf{x}^*, \lambda^*)$ . Moreover, for all fixed  $\lambda > \lambda^*$ , the problems (L1) and (11) have the same value and same set of optimal solutions.*

The proof is in the appendix of the full paper (Zhang et al., 2023a). We will call the smallest possible  $\lambda^*$  the *critical Lagrange multiplier*.

**Proposition 7.3.** *For any fixed value  $\lambda$ , the saddle-point problem (L1) can be expressed as a zero-sum extensive-form game.*

*Proof.* Consider the zero-sum extensive-form game  $\hat{\Gamma}$  between two players, the *mediator* and the *deviator*, with the following structure:

1. Nature picks, with uniform probability, whether or not there is a deviator. If nature picks that there should be a deviator, then nature samples, also uniformly, a deviator  $i \in [n]$ . Nature’s actions are revealed to the deviator, but kept private from the mediator.

<sup>46</sup>Computing optimal equilibria can be phrased as a linear program, and so in principle Adler’s reduction could also lead to an equivalent zero-sum game (Adler, 2013). However, that reduction does not yield an *extensive-form* zero-sum game, which is crucial for our purposes; see Section 7.3.



2. The game  $\Gamma$  is played. All players, except  $i$  if nature picked a deviator, are constrained to according to  $\mathbf{o}_i$ . The deviator plays on behalf of Player  $i$ .
3. Upon reaching terminal node  $z$ , there are two cases. If nature picked a deviator  $i$ , the utility is  $-2\lambda n \cdot u_i(z)$ . If nature did not pick a deviator, the utility is  $2u_M(z) + 2\lambda \sum_{i=1}^n u_i(z)$ .

The mediator’s expected utility in this game is

$$u_M(\boldsymbol{\mu}, \mathbf{o}) - \lambda \sum_{i=1}^n [u_i(\boldsymbol{\mu}, \mathbf{x}_i, \mathbf{o}_{-i}) - u_i(\boldsymbol{\mu}, \mathbf{o})]. \quad \square$$

This characterization enables us to exploit technology used for extensive-form zero-sum game solving to compute optimal equilibria for an entire hierarchy of equilibrium concepts

We will next focus on the computational aspects of solving the induced saddle-point problem (L1) using regret minimization techniques. All of the omitted proofs are deferred to the appendix of the full paper (Zhang et al., 2023a).

The first challenge that arises in the solution of (L1) is that the domain of the minimizing player is unbounded—the Lagrange multiplier is allowed to take any nonnegative value. Nevertheless, we show in the appendix of the full paper (Zhang et al., 2023a) that it suffices to set the Lagrange multiplier to a fixed value (that may depend on the time horizon); appropriately setting that value will allow us to trade off between the equilibrium gap and the optimality gap. We combine this theorem with standard regret minimizers (such as variants of CFR employed in the experiments) to guarantee fast convergence to optimal equilibria.

**Corollary 7.4.** *There exist regret minimization algorithms such that when employed in the saddle-point problem (L1), the average strategy of the mediator  $\bar{\boldsymbol{\mu}} := \frac{1}{T} \sum_{t=1}^T \boldsymbol{\mu}^{(t)}$  converges to the set of optimal equilibria at a rate of  $T^{-1/4}$ . Moreover, the per-iteration complexity is polynomial for communication and certification equilibria (under the nested range condition (Zhang and Sandholm, 2022a)), while for NFCCE, EFCCE and EFCE, implementing each iteration admits a fixed-parameter tractable algorithm.*

Furthermore, we leverage the technique of *optimism*, pioneered by Chiang et al. (2012); Rakhlin and Sridharan (2013b); Syrgkanis et al. (2015), to obtain a faster rate of convergence.

**Corollary 7.5** (Improved rates via optimism). *There exist regret minimization algorithms that guarantee that the average strategy of the mediator  $\bar{\boldsymbol{\mu}} := \frac{1}{T} \sum_{t=1}^T \boldsymbol{\mu}^{(t)}$  converges to the set of optimal equilibria at a rate of  $T^{-1/2}$ . The per-iteration complexity is analogous to Corollary 7.4.*

While this rate is slower than the (near)  $T^{-1}$  rates known for converging to some of those equilibria (Daskalakis et al., 2021; Farina et al., 2022; Piliouras et al., 2022; Anagnostides et al., 2021), Corollaries 7.4 and 7.5 additionally guarantee convergence to *optimal* equilibria; improving the  $T^{-1/2}$  rate of Corollary 7.5 is an interesting direction for future research.

**Last-iterate convergence.** The convergence results we have stated thus far apply for the *average* strategy of the mediator—a typical feature of traditional guarantees in the no-regret framework. Nevertheless, an important advantage of our mediator-augmented formulation is that we can also guarantee *last-iterate convergence* to optimal equilibria in general games. Indeed, this follows readily from our reduction to two-player zero-sum games, leading to the following guarantee.

**Theorem 7.6** (Last-iterate convergence to optimal equilibria in general games). *There exist algorithms that guarantee that the last strategy of the mediator  $\boldsymbol{\mu}^{(T)}$  converges to the set of optimal equilibria at a rate of  $T^{-1/4}$ . The per-iteration complexity is analogous to Corollaries 7.4 and 7.5.*

As such, our mediator-augmented paradigm bypasses known hardness results in the traditional learning paradigm since iterate convergence is no longer tied to Nash equilibria.

### 7.3.2 Thresholding and Binary Search

A significant weakness of the above Lagrangian is that the multiplier  $\lambda^*$  can be large. This means that, in practice, the zero-sum game that needs to be solved to compute an optimal equilibrium could have a large reward range. While this is not a problem for most tabular methods that can achieve high precision, more scalable methods based on reinforcement learning tend to be unable to solve games to the required precision. In this section, we will introduce another Lagrangian-based method for solving the program (11) that will not require solving games with large reward ranges.

Specifically, let  $\tau \in \mathbb{R}$  be a fixed threshold value, and consider the bilinear saddle-point problem

$$\max_{\boldsymbol{\mu} \in \Xi} \min_{\substack{\boldsymbol{\lambda} \in \Delta^{n+1}, \\ \mathbf{x}_i \in \text{co } \mathcal{X}_i : i \in [n]}} \lambda_0 (\mathbf{c}^\top \boldsymbol{\mu} - \tau) - \sum_{i=1}^n \lambda_i \boldsymbol{\mu}^\top \mathbf{A}_i \mathbf{x}_i, \quad (\text{L2})$$

where  $\Delta^k := \{\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^k : \mathbf{1}^\top \boldsymbol{\lambda} = 1\}$  is the probability simplex on  $k$  items. This Lagrangian was also stated—but not analyzed—by Farina et al. (2019b), in the special case of correlated equilibrium concepts (NFCCE, EFCCE, EFCE). Compared to that paper, ours contains a more complete analysis, and is general to more notions of equilibrium.

Like (L1), this Lagrangian is also a zero-sum game, but unlike (L1), the reward range in this Lagrangian is bounded by an absolute constant:

**Proposition 7.7.** *Let  $\Gamma$  be a (mediator-augmented) game in which the reward for all agents is bounded in  $[0, 1]$ . For any fixed  $\tau \in [0, 1]$ , the saddle-point problem (L2) can be expressed as a zero-sum extensive-form game whose reward is bounded in  $[-2, 2]$ .*

*Proof.* Consider the zero-sum extensive-form game  $\hat{\Gamma}$  between two players, the *mediator* and the *deviator*, with the following structure:

1. The deviator picks an index  $i \in [n] \cup \{0\}$ .
2. If  $i \neq 0$ , nature picks whether Player  $i$  can deviate, uniformly at random.
3. The game  $\Gamma$  is played. All players, except  $i$  if  $i \neq 0$  and nature selected that  $i$  can deviate, are constrained to play according to  $\mathbf{o}_i$ . The deviator plays on behalf of Player  $i$ .
4. Upon reaching terminal node  $z$ , there are three cases. If nature picked  $i = 0$ , the utility is  $u_M(z) - \tau$ . Otherwise, if nature picked that Player  $i \neq 0$  can deviate, the utility is  $-2u_i(z)$ . Finally, if nature picked that Player  $i \neq 0$  cannot deviate, the utility is  $2u_i(z)$ .

The mediator’s expected utility in this game is exactly

$$\lambda_M u_M(\boldsymbol{\mu}, \mathbf{o}) - \sum_{i=1}^n \lambda_i [u_i(\boldsymbol{\mu}, \mathbf{x}_i, \mathbf{o}_{-i}) - u_i(\boldsymbol{\mu}, \mathbf{o})]$$

where  $\boldsymbol{\lambda} \in \Delta^{n+1}$  is the deviator’s mixed strategy in the first step. □

The above observations suggest a binary-search-like algorithm for computing optimal equilibria; the pseudocode is given as Algorithm [BinSearch](#). The algorithm solves  $O(\log(1/\epsilon))$  zero-sum games, each to precision  $\epsilon$ . Let  $v^*$  be the optimal value of (11). If  $\tau \leq v^*$ , the value of (L2) is 0, and we will therefore never branch low, in turn implying that  $u \geq v^*$  and  $\ell \geq v^* - \epsilon$ . As a result, we have proven:

**Theorem 7.8.** *Algorithm [BinSearch](#) returns an  $\epsilon$ -approximate equilibrium  $\boldsymbol{\mu}$  whose value to the mediator is at least  $v^* - 2\epsilon$ . If the underlying game solver used to solve (L2) runs in time  $f(\Gamma, \epsilon)$ , then Algorithm [BinSearch](#) runs in time  $O(f(\Gamma, \epsilon) \log(1/\epsilon))$ .*

The differences between the two Lagrangian formulations can be summarized as follows:

---

**Algorithm BinSearch:** Pseudocode for binary search-based algorithm

---

```
1: input: game  $\Gamma$  with mediator reward range  $[0, 1]$ , target precision  $\epsilon > 0$ 
2:  $\ell \leftarrow 0, u \leftarrow 1$ 
3: while  $u - \ell > \epsilon$  do
4:    $\tau \leftarrow (\ell + u)/2$ 
5:   run an algorithm to solve game (L2) until either
6:     (1) it finds a  $\mu$  achieving value  $\geq -\epsilon$  in (L2), or
7:     (2) it proves that the value of (L2) is  $< 0$ 
8:   if case (1) happened then  $\ell \leftarrow \tau$ 
9:   else  $u \leftarrow \tau$ 
10: return the last  $\mu$  found
```

---

1. Using (L1) requires only a single game solve, whereas using (L2) requires  $O(\log(1/\epsilon))$  game solves.
2. Using (L2) requires only an  $O(\epsilon)$ -approximate game solver to guarantee value  $v^* - \epsilon$ , whereas using (L1) would require an  $O(\epsilon/\lambda^*)$ -approximate game solver to guarantee the same, even assuming that the critical Lagrange multiplier  $\lambda^*$  in (L1) is known.

Which is preferred will therefore depend on the application. In practice, if the games are too large to be solved using tabular methods, one can use approximate game solvers based on deep reinforcement learning. In this setting, since reinforcement learning tends to be unable to achieve the high precision required to use (L1), using (L2) should generally be preferred. In Section 8, we back up these claims with concrete experiments.

## 8 Experiments and Conclusion

Here, we describe some of the experiments that we have run using the algorithms described in this part. Since all the techniques in this part are interrelated, this section is standalone rather than a subsection.

### 8.1 Optimal Equilibria in Tabular Games

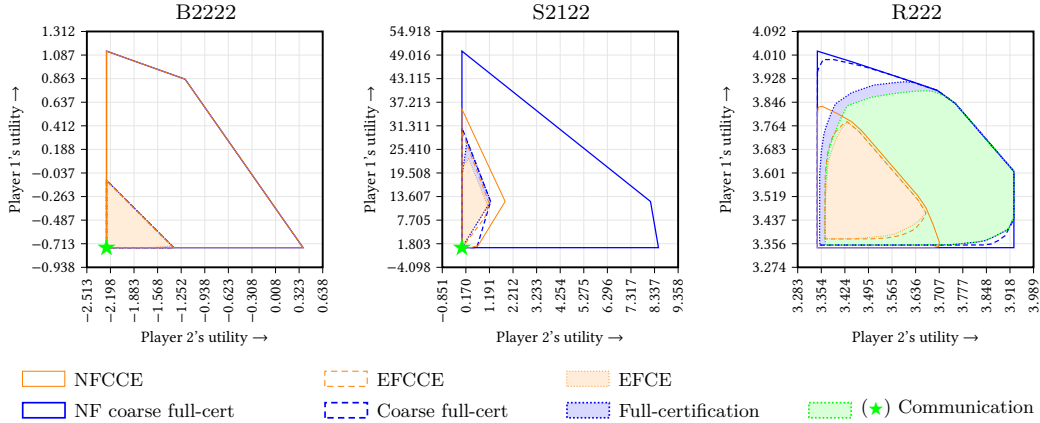
We first extensively evaluate the empirical performance of our two-player zero-sum reduction (Section 7.3.1) for computing seven equilibrium solution concepts across 23 game instances; the results using the method of Section 7.3.2 are slightly inferior, and are included in the appendix of Zhang et al. (2023a).

The game instances we use are also described in detail in the appendix of Zhang et al. (2023a), and belong to following eight different classes of established parametric benchmark games, each identified with an alphabetical mnemonic: **B** – Battleship (Farina et al., 2019b), **D** – Liar’s dice (Lisý et al., 2015), **GL** – Goofspiel (Ross, 1971), **K** – Kuhn poker (Kuhn, 1950b), **L** – Leduc poker (Southey et al., 2005), **RS** – ridesharing game (Zhang et al., 2022b), **S** – Sheriff (Farina et al., 2019b), **TP** – double dummy bridge game (Zhang et al., 2022b).

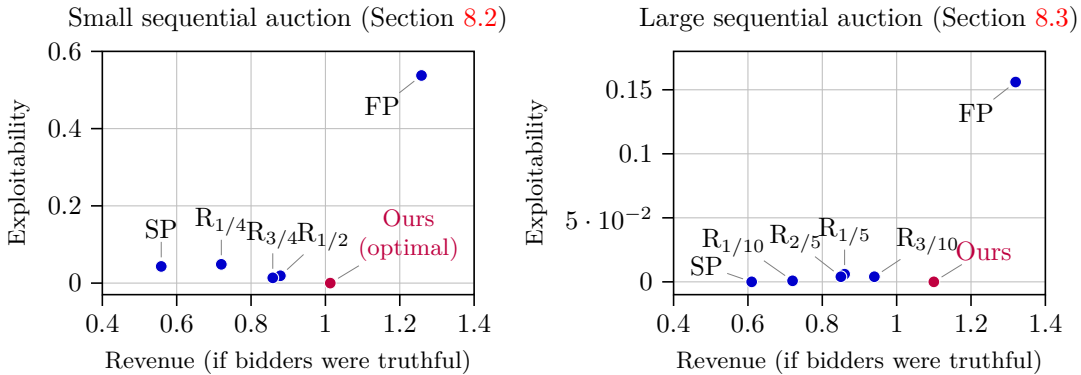
In Figure 19, we have plotted the payoff spaces of some representative instances. The plots show how the polytopes of communication and full-certification equilibria behave relative to correlated equilibria. In the *battleship* and *sheriff* instances, the space of communication equilibrium payoffs is a single point, which implies that the space of NFCE (and hence Nash) equilibrium payoffs is also that single point. Unfortunately, that point is the Pareto-least-optimal point in the space of EFCEs. In the *ridesharing* instances, communication allows higher payoffs. This is because the mediator is allowed to “leak” information between players.

Game	# Nodes	NFCCE		EFCCE		EFCE		COMM		CERT	
		LP	CFR	LP	CFR	LP	CFR	LP	CFR	LP	CFR
B2222	1573	0.00s	0.00s	0.00s	0.01s	0.00s	0.02s	2.00s	1.49s	0.00s	0.02s
B2322	23,839	0.00s	0.01s	3.00s	0.69s	9.00s	1.60s	timeout	4m 41s	2.00s	1.24s
B2323	254,239	6.00s	0.33s	1m 21s	14.23s	3m 40s	44.87s	timeout	timeout	37.00s	40.45s
B2324	1,420,639	38.00s	2.73s	timeout	3m 1s	timeout	10m 48s	timeout	timeout	timeout	6m 14s
D32	1017	0.00s	0.01s	0.00s	0.02s	12.00s	0.40s	0.00s	0.06s	0.00s	0.01s
D33	27,622	2m 17s	12.93s	timeout	1m 46s	timeout	timeout	timeout	4m 37s	4.00s	3.14s
GL3	7735	0.00s	0.01s	1.00s	0.02s	0.00s	0.01s	timeout	7.72s	0.00s	0.02s
K35	1501	49.00s	0.76s	46.00s	0.67s	57.00s	0.55s	1.00s	0.03s	0.00s	0.01s
L3132	8917	26.00s	0.59s	8m 43s	5.13s	8m 18s	6.10s	8.00s	3.46s	1.00s	0.10s
L3133	12,688	38.00s	0.94s	20m 26s	8.88s	21m 25s	6.84s	12.00s	3.40s	1.00s	0.22s
L3151	19,981	timeout	15.12s	timeout	timeout	timeout	timeout	timeout	16.73s	2.00s	0.21s
L3223	15,659	4.00s	0.44s	1m 10s	2.94s	2m 2s	5.52s	19.00s	18.19s	1.00s	0.61s
L3523	1,299,005	timeout	1m 7s	timeout	timeout	timeout	timeout	timeout	timeout	timeout	2m 58s
S2122	705	0.00s	0.00s	0.00s	0.01s	0.00s	0.02s	2.00s	0.35s	0.00s	0.02s
S2123	4269	0.00s	0.01s	1.00s	0.06s	1.00s	0.15s	1m 33s	59.63s	1.00s	0.15s
S2133	9648	1.00s	0.02s	3.00s	0.11s	3.00s	0.49s	timeout	12m 11s	2.00s	0.92s
S2254	712,552	1m 58s	7.43s	timeout	22.01s	timeout	3m 34s	timeout	timeout	timeout	2m 42s
S2264	1,303,177	3m 43s	11.74s	timeout	39.23s	timeout	timeout	timeout	timeout	timeout	timeout
TP3	910,737	1m 38s	7.44s	timeout	13.76s	timeout	13.46s	timeout	timeout	timeout	26.70s
RS212	598	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s	2.00s	0.01s	0.00s	0.00s
RS222	734	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s	3.00s	0.01s	0.00s	0.00s
RS213	6274	timeout	14.68s	timeout	15.54s	timeout	23.37s	6m 25s	8.74s	0.00s	0.02s
RS223	6238	timeout	timeout	timeout	timeout	timeout	timeout	8m 54s	4.00s	1.00s	0.01s

**Table 18:** Experimental comparison between our learning-based approach (‘CFR’, Section 7.3.1) and our linear-programming-based method (‘LP’, Section 6.4.1 and Section 5.3.1). Within each pair of cells corresponding to ‘LP’ vs ‘CFR,’ the faster algorithm is shaded blue while the hue of the slower algorithm depends on how much slower it is. If both algorithms timed out, they are both shaded gray.



**Figure 19:** Payoff spaces for various games and notions of equilibrium. The symbol  $\star$  indicates that the set of communication equilibrium payoffs for that game is (at least, modulo numerical precision) that single point. In the battleship instance, many of the notions overlap.



FP: First-price auction    SP: Second-price auction     $R_p$ : Second-price action with reserve price  $p$

**Figure 20:** Exploitability is measured by summing the best response for both bidders to the mechanism. Zero exploitability corresponds to incentive compatibility. In a sequential auction with budgets, our method is able to achieve higher revenue than second-price auctions and better incentive compatibility than a first-price auction.

## 8.2 Exact Sequential Auction Design

Next, we use our approach to derive the optimal mechanism for a sequential auction design problem. In particular, we consider a two-round auction with two bidders, each starting with a budget of 1. The valuation for each item for each bidder is sampled uniformly at random from the set  $\{0, 1/4, 1/2, 3/4, 1\}$ . We consider a mediator-augmented game in which the principal chooses an outcome (allocation and payment for each player) given their reports (bids). We use CFR+ (Tammelin et al., 2015) as learning algorithm and a fixed Lagrange multiplier  $\lambda := 25$  to compute the optimal communication equilibrium that corresponds to the optimal mechanism. We terminated the learning procedure after 10000 iterations, at a duality gap for (L1) of approximately  $4.2 \times 10^{-4}$ . Figure 20 (left) summarizes our results. On the y-axis we show how exploitable (that is, how incentive-incompatible) each of the considered mechanisms are, confirming that for this type of sequential settings, second-price auctions (SP) with or without reserve price, as well as the first-price auction (FP), are typically incentive-incompatible. On the x-axis, we report the hypothetical revenue that the mechanism would extract assuming truthful bidding. Our mechanism is provably incentive-compatible and extracts a larger revenue than all considered second-price mechanisms. It also would extract less revenue than the hypothetical first-price auction if the bidders behaved truthfully (of course, real bidders would not behave honestly in the first-price auction but

rather would shade their bids downward, so the shown revenue benchmark in Figure 20 is actually not achievable). Intriguingly, we observed that 8% of the time the mechanism gives an item away for free. Despite appearing irrational, this behavior can incentivize bidders to use their budget earlier in order to encourage competitive bidding, and has been independently discovered in manual mechanism design recently (Deng et al., 2021).

### 8.3 Scalable Sequential Auction Design via Deep Reinforcement Learning

We also combine our framework with deep-learning-based algorithms for scalable equilibrium computation in two-player zero-sum games to compute optimal mechanisms in two sequential auction settings. To compute an optimal mechanism using our framework, we use the PSRO algorithm (Lanctot et al., 2017), a deep reinforcement learning method based on the double oracle algorithm that has empirically scaled to large games such as Starcraft (Vinyals et al., 2019) and Stratego (McAleer et al., 2020), as the game solver in Algorithm BinSearch.<sup>47</sup> To train the best responses, we use proximal policy optimization (PPO) (Schulman et al., 2017).

First, to verify that the deep learning method is effective, we replicate the results of the tabular experiments in Section 8.2. We find that PSRO achieves the same best response values and optimal equilibrium value computed by the tabular experiment, up to a small error. These results give us confidence that our method is correct.

Second, to demonstrate scalability, we run our deep learning-based algorithm on a larger auction environment that would be too big to solve with tabular methods. In this environment, there are four rounds, and in each round the valuation of each player is sampled uniformly from  $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ . The starting budget of each player is, again, 1. We find that, like the smaller setting, the optimal revenue of the mediator is  $\approx 1.1$  (right-side of Figure 20). This revenue exceeds the revenue of every second-price auction (none of which have revenue greater than 1).<sup>48</sup>

### 8.4 Conclusion

We proposed a new paradigm of learning in games. It applies to mechanism design, information design, and solution concepts in multi-player extensive-form games such as correlated, communication, and certification equilibria. Leveraging a Lagrangian relaxation, our paradigm reduces the problem of computing optimal equilibria to determining minimax equilibria in zero-sum extensive-form games. We also demonstrated the scalability of our approach for *computing* optimal equilibria by attaining state-of-the-art performance in benchmark tabular games, and by solving a sequential auction design problem using deep reinforcement learning. Along the way, we have shown parameterized complexity results—both upper and lower bounds—for the special case of computing optimal *correlated* equilibria.

Possible directions of future research include the following.

1. Is there a better-than-quadratic-size linear program or similar algorithm for communication equilibria?
2. Is it possible to extend our augmented game construction to also cover *normal-form* correlated equilibria while maintaining efficiency?
3. Investigate further the comparison between communication and correlation in games. For example, when and why do communication equilibria achieve higher social welfare than extensive-form correlated equilibria?
4. Extend CorrelationDAG in such a way that it also has polynomial size in all triangle-free games.

<sup>47</sup>We also tested PSRO on the Lagrangian (L1), but this proved to be incompatible with deep learning due to the large reward range induced by the multiplier  $\lambda$ .

<sup>48</sup>We are inherently limited in this setting by the inexactness of best responses based on deep reinforcement learning; as such, it is possible that these values are not exact. However, because of the success of above tabular experiment replications, we believe that our results should be reasonably accurate.

5. An intelligent combination—rather than merely a selection of one versus the other—of the correlation DAG and the column generation algorithm may lead to faster practical algorithms.
6. Investigate possible use of the payoff structure in the game; for example, investigate extensions of the concept of *smooth games* ([Roughgarden, 2015](#)).

# Part III

# Learning in Games

## 9 Preliminaries

Before proceeding, we must introduce some notation and background that will be fundamental in this part.

### 9.1 $\Phi$ -Regret Minimization

In the framework of online learning, a learner interacts with an adversary over a sequence of rounds. In each round, the learner selects a strategy, whereupon the adversary constructs a utility function which is subsequently observed by the learner. Throughout this paper, we allow the adversary to be strongly adaptive, so that the utility function at the  $t$ th round  $u^{(t)} : \mathcal{X} \ni \mathbf{x} \mapsto \langle \mathbf{u}^{(t)}, \mathbf{x} \rangle$  can depend on the strategy of the learner at that round. We assume that utilities belong to  $\mathcal{U} := \{\mathbf{u} : |\langle \mathbf{u}, \mathbf{x} \rangle| \leq 1 \forall \mathbf{x} \in \mathcal{X}\}$ . It will be convenient to use  $\|\mathbf{x}\|_{\mathcal{X}} := \max_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{u}, \mathbf{x} \rangle$  for the induced norm.

We measure the performance of an online learning algorithm as follows. Suppose that  $\Phi \subseteq (\text{co } \mathcal{X})^{\mathcal{X}}$  is a set of deviations. If the learner outputs in each round a *mixed strategy*  $\pi^{(t)} \in \Delta(\mathcal{X})$ , its (time-average)  $\Phi$ -regret (Greenwald and Hall, 2003; Stoltz and Lugosi, 2007) is defined as

$$\text{REG}_{\Phi}^T := \frac{1}{T} \max_{\phi \in \Phi} \sum_{t=1}^T \left\langle \mathbf{u}^{(t)}, \mathbb{E}_{\mathbf{x}^{(t)} \sim \pi^{(t)}} [\phi(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}] \right\rangle. \quad (12)$$

In the special case where  $\Phi$  contains only *constant transformations*, one recovers the notion of *external regret*. On the other extreme, *swap regret* corresponds to  $\Phi$  containing all functions  $\mathcal{X} \rightarrow \mathcal{X}$ .

It is sometimes assumed that the learner instead selects in each round a strategy  $\mathbf{x}^{(t)} \in \text{co } \mathcal{X}$ . To translate (12) in that case, we introduce the *extended mapping* of a deviation  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  as  $\phi^{\delta} := \mathbb{E}_{\mathbf{x}' \sim \delta(\mathbf{x})} [\phi(\mathbf{x}')]$ , where  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  is a function that is *consistent* in the sense that  $\mathbb{E}_{\mathbf{x}' \sim \delta(\mathbf{x})} [\mathbf{x}'] = \mathbf{x}$ . A canonical example of such a function  $\delta$  is the *behavioral strategy map*  $\beta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$ , which returns the unique (ignoring actions at decision points reached with probability zero) mixed strategy whose actions at different decision points are independent and whose expectation is  $\mathbf{x}$ . We give another example of a consistent map in Section 11.5.2. Accordingly, we let  $\Phi^{\delta}$  denote all extended mappings. In this context,  $\Phi^{\delta}$ -regret is defined as

$$\text{REG}_{\Phi^{\delta}}^T := \frac{1}{T} \max_{\phi^{\delta} \in \Phi^{\delta}} \sum_{t=1}^T \left\langle \mathbf{u}^{(t)}, \phi^{\delta}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)} \right\rangle.$$

We are interested in algorithms whose regret is bounded by  $\epsilon$  after  $T = \text{poly}(N, 1/\epsilon)$  rounds. We refer to such algorithms as *fully-polynomial no-regret learners*.

**Remark 9.1** (Swap versus internal regret). When it comes to defining correlated equilibria in normal-form games, there are two prevalent definitions appearing in the literature; one is based on *internal regret*, while the other on *swap regret* (e.g., (Ganor and Karthik C. S., 2018; Goldberg and Roth, 2016)). The key difference is that internal regret only contains deviations that swap a *single action*—thereby being weaker. Nevertheless, it is not hard to see that swap regret can only be larger by a factor of  $|\mathcal{X}|$  (Blum and Mansour, 2007), where we recall that  $\mathcal{X}$  denotes the set of pure strategies. So, in normal-form games those two definitions are polynomially equivalent, and in most applications one can safely switch from one to the other.

However, this is certainly not the case in games with an exponentially large action space, such as extensive-form games. In fact, the definition of internal regret itself is problematic when the action set is exponentially large:



the uniform distribution always attains an error of at most  $1/|\mathcal{X}|$ . Consequently, any guarantee for  $\epsilon \geq 1/|\mathcal{X}|$  is vacuous. That is, if  $|\mathcal{X}|$  is exponentially large, an algorithm that requires a number of iterations polynomial in  $1/\epsilon$ —which is what we expect to get from typical no-regret dynamics—would need an exponential number of iterations to yield a non-trivial guarantee; this issue with internal regret was also observed by Fujii (2023). Nevertheless, internal regret in the context of games with an exponentially large action set was used in a recent work by Chen et al. (2023), who provided oracle-efficient algorithms for minimizing internal regret.

## 9.2 The GGM Construction

Gordon et al. (2008), building on earlier work by Blum and Mansour (2007) and Stoltz and Lugosi (2005), came up with a general recipe for minimizing  $\Phi^\delta$ -regret. That construction relies on a no-regret learning algorithm on the set of deviations  $\Phi^\delta$ , which we denote by  $\mathcal{R}_\Phi$ . Then, a  $\Phi^\delta$ -regret minimizer on  $\text{co } \mathcal{X}$  can be constructed as follows: on each iteration  $t = 1, \dots, T$ , the learner performs the following steps.

1. Receive  $\phi^{(t)}$  from  $\mathcal{R}_\Phi$ . Select  $\mathbf{x}^{(t)} \in \text{co } \mathcal{X}$  as an  $\epsilon$ -fixed point of  $\phi^{(t)}$ :  $\|\phi^{(t)}(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}\|_{\mathcal{X}} \leq \epsilon$ .
2. Upon receiving utility  $\mathbf{u}^{(t)} \in \mathcal{U}$ , pass utility  $\Phi^\delta \ni \phi^\delta \mapsto \langle \mathbf{u}^{(t)}, \phi^\delta(\mathbf{x}^{(t)}) \rangle$  to  $\mathcal{R}_\Phi$ .

**Theorem 9.2** (Gordon et al., 2008). *Suppose that  $\text{REG}^T$  is the external regret incurred by  $\mathcal{R}_\Phi$ . After  $T$  rounds of the above algorithm, we have*

$$\max_{\phi^\delta \in \Phi^\delta} \frac{1}{T} \sum_{t=1}^T \langle \mathbf{u}^{(t)}, \phi^\delta(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)} \rangle \leq \text{REG}^T + \epsilon.$$

In Section 11.5.1, we will relax the requirement of needing (approximate) fixed points, while at the same time maintaining the guarantee of Theorem 9.2.

## 9.3 Convergence to Correlated Equilibria

Notions of  $\Phi$ -regret correspond naturally to notions of correlated equilibria. Therefore, our results also have implications for no-regret learning algorithms that converge to correlated equilibria. Here, we formalize this connection. Consider an  $n$ -player game in which player  $i$ 's strategy set is a tree-form strategy set  $\mathcal{X}_i$ , and player  $i$ 's utility is given by a multilinear map  $u_i : \mathcal{X}_1 \times \dots \times \mathcal{X}_n \rightarrow [-1, 1]$ . For each player  $i$ , let  $\Phi_i \subseteq (\text{co } \mathcal{X}_i)^{\mathcal{X}_i}$  be a set of deviations for player  $i$ . Finally let  $\Phi = (\Phi_1, \dots, \Phi_n)$ .

**Definition 9.3.** A distribution  $\pi \in \Delta(\mathcal{X}_1 \times \dots \times \mathcal{X}_n)$  is called a *correlated profile*. A correlated profile  $\pi$  is an  $\epsilon$ - $\Phi$ -*equilibrium* if no player  $i$  can profit more than  $\epsilon$  via any of the deviations  $\phi_i \in \Phi_i$  to its strategy. That is,  $\mathbb{E}_{\mathbf{x} \sim \pi} u_i(\phi_i(\mathbf{x}_i), \mathbf{x}_{-i}) \leq \mathbb{E}_{\mathbf{x} \sim \pi} u_i(\mathbf{x}_i, \mathbf{x}_{-i}) + \epsilon$  for all players  $i$  and  $\phi_i \in \Phi_i$ .

For example, we can define *k-mediator equilibria* and *degree-k swap equilibria* by setting  $\Phi_i$  to  $\Phi_{\text{med}}^k$  and  $\Phi_{\text{poly}}^k$ , respectively. The following celebrated result follows immediately from the definitions of equilibrium and regret.

**Proposition 9.4.** *Suppose that every player  $i$  plays according to a regret minimizer whose  $\Phi_i$ -regret is at most  $\epsilon$  after  $T$  rounds. Let  $\pi_i^{(t)} \in \Delta(\mathcal{X}_i)$  be the distribution played by player  $i$  at round  $t$ . Let  $\pi^{(t)} \in \Delta(\mathcal{X}_1) \times \dots \times \Delta(\mathcal{X}_n)$  be the product distribution whose marginal on  $\mathcal{X}_i$  is  $\pi_i^{(t)}$ . Then the average strategy profile, that is, the distribution  $\frac{1}{T} \sum_{t \in [T]} \pi^{(t)}$ , is an  $\epsilon$ - $\Phi$ -equilibrium.*

Some common choices of  $\Phi$ , and corresponding equilibrium notions, are in Table 21.

Finally, notationally, it will be convenient for us to denote  $|\Sigma| = N$ .

Deviations $\Phi$	Equilibrium concept	References
Constant (external), $\Phi = \{\phi : \mathbf{x} \mapsto \mathbf{x}_0 \mid \mathbf{x}_0 \in \mathcal{X}\}$	Normal-form coarse correlated	Moulin and Vial (1978)
Trigger (see Section 10.2)	Extensive-form correlated	von Stengel and Forges (2008)
Communication (see Section 10.2)	Communication	Forges (1986); Myerson (1986)
Linear / Untimed communication	Linear correlated	Farina and Papis (2023); Section 10
Swap, $\Phi = \mathcal{X}^{\mathcal{X}}$	Normal-form correlated	Aumann (1974)

**Table 21:** Some examples of deviation sets  $\Phi$  and corresponding notions of correlated equilibrium, in increasing order of size of  $\Phi$  (and thus increasing tightness of the equilibrium concept)

## 10 Mediator Interpretation and Faster Learning Algorithms for Linear Correlated Equilibria

### 10.1 Introduction

In this paper, we consider a notion of regret first studied for extensive-form games by Farina and Papis (2023), namely, regret with respect to the set of *linear functions* from the strategy set to itself. This notion is a natural stepping stone between external regret, which is very well studied, and swap regret, for which achieving  $\text{poly}(N) \cdot T^c$  regret, where  $N$  is the size of the decision problem and  $c < 1$ , is a long-standing open problem. We make two main contributions.

The first contribution is conceptual: we give, for extensive-form games, an *interpretation* of the set of linear deviations. More specifically, we will first introduce a set of deviations, which we will call the *untimed communication (UTC) deviations* that, a priori, seems very different from the set of linear deviations at least on a conceptual level. The deviation set, rather than being defined *algebraically* (linear functions), will be defined in terms of an interaction between a *deviator*, who wishes to evaluate the deviation function at a particular input, and a *mediator*, who answers queries about the input. We will show the following result, which is our first main theorem:

**Theorem 10.1.** *The untimed communication deviations are precisely the linear deviations.*

The mediator-based framework is more in line with other extensive-form deviation sets—indeed, all prior notions of regret for extensive form, to our knowledge, including all the notions discussed above, can be expressed in terms of the framework. As such, the above theorem places linear deviations firmly within the same framework usually used to study deviations in extensive form.

We will then demonstrate that the set of UTC deviations is expressible in terms of *scaled extensions* (Farina et al., 2019c), opening up access to a wide range of extremely fast algorithms for regret minimization, both theoretically and practically, for UTC deviations and thus also for linear deviations. Our second main theorem is as follows.

**Theorem 10.2** (Faster linear-swap regret minimization). *There exists a regret minimizer with regret  $O(N^2\sqrt{T})$  against all linear deviations, and whose per-iteration complexity is dominated by the complexity of computing a fixed point of a linear map  $\phi^{(t)} : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ .*

In particular, using the algorithm of Cohen et al. (2021) to solve the linear program of finding a fixed point, our per-iteration complexity is  $\tilde{O}(N^\omega)$ , where  $\omega \approx 2.37$  is the current matrix multiplication constant and  $\tilde{O}$  hides logarithmic factors. We elaborate on the fixed-point computation in Section 10.5. This improves substantially on the result of Farina and Papis (2023), which has the same regret bound but whose per-iteration computation involved a *quadratic* program (namely, an  $\ell_2$  projection), which has higher complexity than a linear program (they give a bound of  $\tilde{O}(N^{10})$ ). Finally, we demonstrate via experiments that our method is also empirically faster than the prior method.

## 10.2 Mediators and UTC Deviations

For extensive-form games, linear-swap regret was recently studied in detail by [Farina and Pipis \(2023\)](#): they provide a characterization of the set  $\Phi_{\text{LIN}}$  when  $\mathcal{X}$  is a sequence-form polytope, and thus derive an algorithm for minimizing  $\Phi_{\text{LIN}}$ -regret over  $\mathcal{X}$ . Their paper is the starting point of ours.

With the notable exception of linear deviations, most sets of deviations  $\Phi$  for extensive-form games are defined by interactions between a *mediator* who holds a strategy  $\mathbf{x} \in \mathcal{X}$ , and a *deviator*, who should compute the function  $\phi(\mathbf{x})$  by making queries to the mediator. The set of deviations is then defined by what queries that the player is allowed to make. Before continuing, we will first formulate the sets  $\Phi$  mentioned in [Section 9.3](#) in this paradigm, for intuition. For a given decision point  $j$ , call an action  $a \in A_j$  the *recommended action at  $j$* , denoted  $a(\mathbf{x}, j)$ , if  $\mathbf{x}[ja] = 1$ . Since  $\mathbf{x}$  is a sequence-form strategy, it is possible for a decision point to have no recommended action if its parent  $p_j$  is itself not recommended.

- Constant (NFCCE): The deviator cannot to make any queries to the mediator.
- Trigger (EFCE): The deviator, upon reaching a decision point  $j$ , learns the recommended action (if any) at  $j$  before selecting its own action.
- Communication: The deviator maintains a *state* with the mediator, which is a sequence  $\sigma$ , initially  $\emptyset$ . Upon reaching a decision point  $j$ , the deviator selects a decision point  $j' \in C_\sigma$  (possibly  $j' \neq j$ ) at which to query the mediator, the deviator observes the recommendation  $a' = a(\mathbf{x}, j')$ , then the deviator must pick an action  $a \in A_j$ . The state is updated to  $j'a'$ .
- Swap (NFCE): The deviator learns the whole strategy  $\mathbf{x}$  before selecting its strategy.

An example of a communication deviation can be found in [Section 10.4](#). Of these, the closest notion to ours is the notion of communication deviation, and that is the starting point of our construction. One critical property of communication deviations is that the mediator and deviator “share a clock”: for every decision point reached, the deviator must make exactly one query to the mediator. As the name suggests, our set of *untimed* deviations results from removing this timing restriction, and therefore allowing the deviator to make *any number* (zero, one, or more than one) of queries to the mediator for every decision point reached. We formally define the decision problem faced by an untimed deviator as follows.

**Definition 10.3.** The *UTC decision problem* corresponding to a given tree-form decision problem is defined as follows. Nodes are identified with pairs  $(s, \tilde{s})$  where  $s, \tilde{s} \in \Sigma \cup \mathcal{J}$ .  $s$  represents the state of the real decision problem, and  $\tilde{s}$  represents the state of the mediator. The root is  $(\emptyset, \emptyset) \in \Sigma \times \Sigma$ .

1.  $(\sigma, \tilde{\sigma}) \in \Sigma \times \Sigma$  is an observation point. The deviator observes the next decision point  $j \in C_\sigma$ , and the resulting decision point is  $(j, \tilde{\sigma})$
2.  $(j, \tilde{j}) \in \mathcal{J} \times \mathcal{J}$  is an observation point. The deviator observes the recommendation  $a = a(\mathbf{x}, \tilde{j})$ , and the resulting decision point is  $(j, \tilde{j}a)$ .
3.  $(j, \tilde{\sigma}) \in \mathcal{J} \times \Sigma$  is a decision point. The deviator can choose to either play an action  $a \in A_j$ , or to query a decision point  $\tilde{j} \in C_{\tilde{\sigma}}$ . In the former case, the resulting observation point is  $(ja, \tilde{\sigma})$  for  $a \in A_j$ ; in the latter case, the resulting observation point is  $(j, \tilde{j})$ .

Any mixed strategy of the deviator in this decision problem defines a function  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , where  $\phi(\mathbf{x})[\sigma]$  is the probability that an untimed deviator plays all the actions on the path to  $\sigma$  when the mediator recommends according to pure strategy  $\mathbf{x}$ . We thus define:

**Definition 10.4.** An *UTC deviation* is any function  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  induced by a mixed strategy of the deviator in the UTC decision problem.

Clearly, the set of UTC deviations is at least as large as the set of communication deviations, and at most as large as the set of swap deviations. In the next section, we will discuss how to represent UTC deviations, and show that UTC deviations coincide precisely with linear deviations.

### 10.3 Representation of UTC Deviations and Equivalence between UTC and Linear Deviations

Since UTC deviations are defined by a decision problem, one method of representing such deviations is to express it as a tree-form decision problem and use the sequence-form representation. However, the UTC decision problem is not a tree—it is a DAG, since there are multiple ways of reaching any given decision point  $(j, \bar{\sigma})$  depending on the ordering of the player’s past actions and queries. Converting it to a tree by considering the tree of paths through the DAG would result in an exponential blowup: a decision point  $(j, \bar{\sigma})$ , where  $j$  is at depth  $k$  and  $\bar{\sigma}$  is at depth  $\ell$ , can be reached in roughly  $\binom{k+\ell}{k}$  ways, so the total number of paths can be exponential in the depth of the decision problem even when the number of sequences,  $N = |\Sigma|$ , is not.

However, it is still possible to define the “sequence form” of a pure deviation in our UTC decision problem as follows<sup>49</sup>: it is a pair of matrices  $(\mathbf{A}, \mathbf{B})$  where  $\mathbf{A} \in \{0, 1\}^{\Sigma \times \Sigma}$  encodes the part corresponding to sequences  $(\sigma, \bar{\sigma})$ , and  $\mathbf{B} \in \{0, 1\}^{\mathcal{J} \times \mathcal{J}}$  encodes the part corresponding to decision points  $(j, \bar{j})$ .  $\mathbf{A}(\sigma, \bar{\sigma}) = 1$  if the deviator plays all the actions on *some* path to observation point  $(\sigma, \bar{\sigma})$ , and similarly  $\mathbf{B}(j, \bar{j}) = 1$  if the deviator plays all the actions on some path to observation node  $(j, \bar{j})$ . Since the only possible way for two paths to end at the same observation point is for the deviator to have changed the order of actions and queries, for any given pure strategy of the deviator, at most one path can exist for both cases. Therefore, the set of mixed sequence-form deviations can be expressed using the following set of constraints:

$$\begin{aligned} \mathbf{A}[p_j, \bar{\sigma}] + \mathbf{B}[j, p_{\bar{\sigma}}] &= \sum_{a \in A_j} \mathbf{A}[ja, \bar{\sigma}] + \sum_{\bar{j} \in C_{\bar{\sigma}}} \mathbf{B}[j, \bar{j}] & \forall j \in \mathcal{J}, \bar{\sigma} \in \Sigma \\ \mathbf{A}[\emptyset, \emptyset] &= 1 \\ \mathbf{A}[\emptyset, \bar{\sigma}] &= 0 & \forall \bar{\sigma} \neq \emptyset \\ \mathbf{A}, \mathbf{B} &\geq 0 \end{aligned}$$

where, in a slight abuse of notation, we define  $\mathbf{B}[j, p_{\emptyset}] := 0$  for every  $j \in \mathcal{J}$ . Moreover, for any pair of matrices  $(\mathbf{A}, \mathbf{B})$  satisfying the constraint system and therefore defining some deviation  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , it is easy to compute how  $\phi$  acts on any  $\mathbf{x} \in \mathcal{X}$ : the probability that the deviator plays all the actions on the  $\emptyset \rightarrow \sigma$  path is simply given by

$$\sum_{\bar{\sigma} \in \Sigma} \mathbf{x}[\bar{\sigma}] \mathbf{A}[\sigma, \bar{\sigma}] = (\mathbf{A}\mathbf{x})[\sigma],$$

and therefore  $\phi$  is nothing more than a matrix multiplication with  $\mathbf{A}$ , that is,  $\phi(\mathbf{x}) = \mathbf{A}\mathbf{x}$ . We have thus shown that every UTC deviation is linear, that is,  $\Phi_{\text{UTC}} \subseteq \Phi_{\text{LIN}}$ . In fact, the reverse inclusion holds too:

**Theorem 10.5.** *The UTC deviations are precisely the linear deviations. That is,  $\Phi_{\text{UTC}} = \Phi_{\text{LIN}}$ .*

The proof is deferred to the appendix of the full paper (Zhang et al., 2024d). Since the two sets are equivalent, in the remainder of the paper, we will use the terms *UTC deviation* and *linear deviation* (similarly, *UTC regret* and *linear-swap regret*) interchangeably.

<sup>49</sup>This construction is a special case of the more general construction of sequence forms for DAG decision problems explored by Zhang et al. (2023b) in the case of team games.

## 10.4 Example

In this section, we provide an example in which the UTC deviations are strictly more expressive than the communication deviations. Consider the game in Figure 22. The subgames rooted at **D** and **E** are guessing games, where  $\blacktriangle$  must guess  $\blacktriangledown$ 's action, with a large penalty for guessing wrong. Consider the correlated profile that mixes uniformly among the four pure profiles  $(\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_1, \mathbf{f}_i, \mathbf{g}_j)$  for  $i, j \in \{1, 2\}$ . In this profile, the information that  $\blacktriangle$  needs to guess perfectly is contained in the recommendations: the recommendation at **A** tells it how to guess at **D**, and the recommendation at **B** tells it how to guess at **E**. With a communication deviation,  $\blacktriangle$  cannot access this information in a profitable way, since upon reaching **C**,  $\blacktriangle$  must immediately make its first mediator query. Hence, this profile is a communication equilibrium. However, with an *untimed* communication deviation,  $\blacktriangle$  can profit: it should, upon reaching<sup>50</sup> **C**, play action  $\mathbf{c}_2$  *without making a mediator query*, and then query **A** if it observes **D**, and **B** if it observes **E**. This deviation is allowed only due to the untimed nature of UTC deviations allows the deviating player to *delay* its query to the mediator until it reaches either **D** or **E**. In a *timed* communication deviation, this deviation is impossible, because the player must make its first query (**A**, **B**, or **C**) *before* reaching **D** or **E**, and thus that query cannot be conditioned on which one of **D** or **E** will be reached.

Another example, where the player can profit from making *more than one* query, and untimed deviations affects the set of possible equilibrium outcomes, can be found in the appendix of the full paper (Zhang et al., 2024d).

## 10.5 Regret Minimization on $\Phi_{\text{UTC}}$

In this section, we discuss how Theorem 10.5 can be used to construct very efficient  $\Phi_{\text{LIN}}$ -regret minimizers, both in theory and in practice. The key observation we use here is due to Zhang et al. (2023b): they observed that DAG decision problems have a structure that allows them to be expressed as *scaled extensions*, allowing the application of the *counterfactual regret minimization* (CFR) framework (Zinkevich et al., 2007; Farina et al., 2019a):

**Theorem 10.6** (CFR for  $\Phi_{\text{LIN}}$ , special case of Zhang et al., 2023b). *CFR-based algorithms can be used to construct an external regret minimizer on  $\Phi_{\text{UTC}}$  (and thus also on  $\Phi_{\text{LIN}}$ ) with  $O(N^2\sqrt{T})$  regret and  $O(N^2)$  per-iteration complexity.*

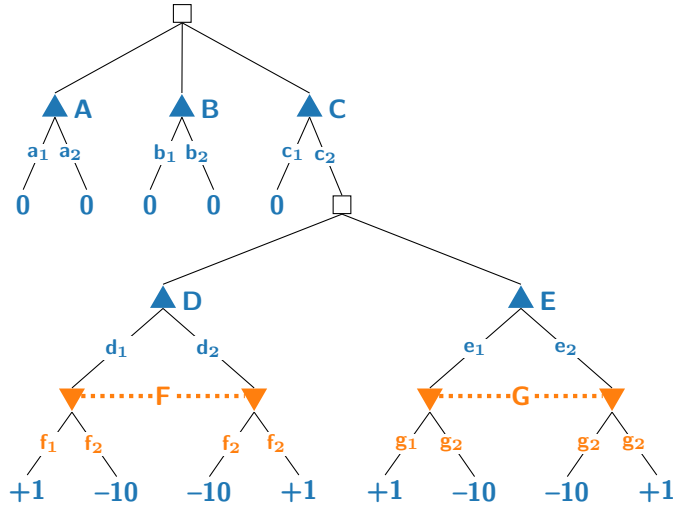
Applying Theorem 9.2 now yields:

**Theorem 10.7.** *CFR-based algorithms can be used to construct a  $\Phi_{\text{LIN}}$ -regret minimizer with  $O(N^2\sqrt{T})$  regret, and per-iteration complexity dominated by the complexity of computing a fixed point of a linear transformation  $\phi^{(t)} : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ .*

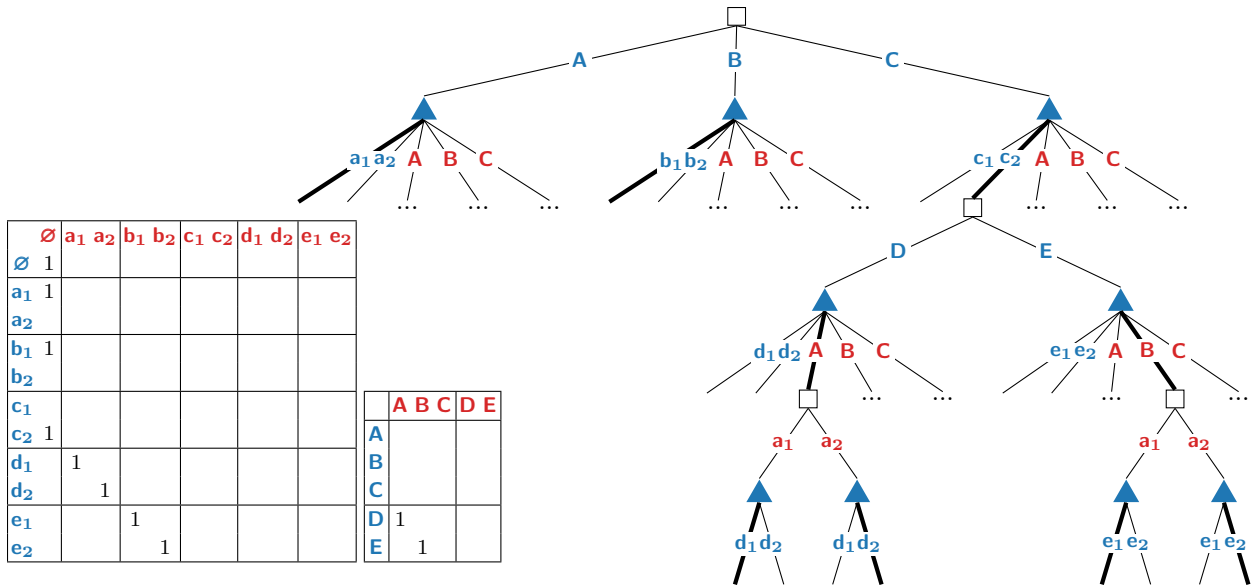
As mentioned in the introduction, this significantly improves the per-iteration complexity of linear-swap regret minimization. Fixed points can be computed by finding a feasible solution to the constraint system  $\{\mathbf{x} \in \text{co } \mathcal{X}, \mathbf{A}\mathbf{x} = \mathbf{x}\}$ , where  $\mathbf{x} \in \text{co } \mathcal{X}$  is expressed using the sequence-form constraints. This is a linear program with  $O(N)$  variables and constraints, so the LP algorithm of Cohen et al. (2021) yields a fixed-point computation algorithm with runtime  $\tilde{O}(N^\omega)$ .

For comparison, the algorithm of Farina and Papis (2023) requires an  $\ell_2$  projection onto  $\mathcal{X}$  on every iteration, which requires solving a convex quadratic program; the authors of that paper derive a bound of  $\tilde{O}(N^{10})$ , which, although polynomial, is much slower than our algorithm. CFR-based algorithms are currently the fastest practical regret minimizers (Brown and Sandholm, 2019a; Farina et al., 2021a)—therefore, showing that our method allows such algorithms to be applied is also a significant practical step. In Section 10.7, we will show empirically that the resulting algorithm is significantly better than the previously-known state of the art, in terms of both per-iteration time complexity and number of iterations.

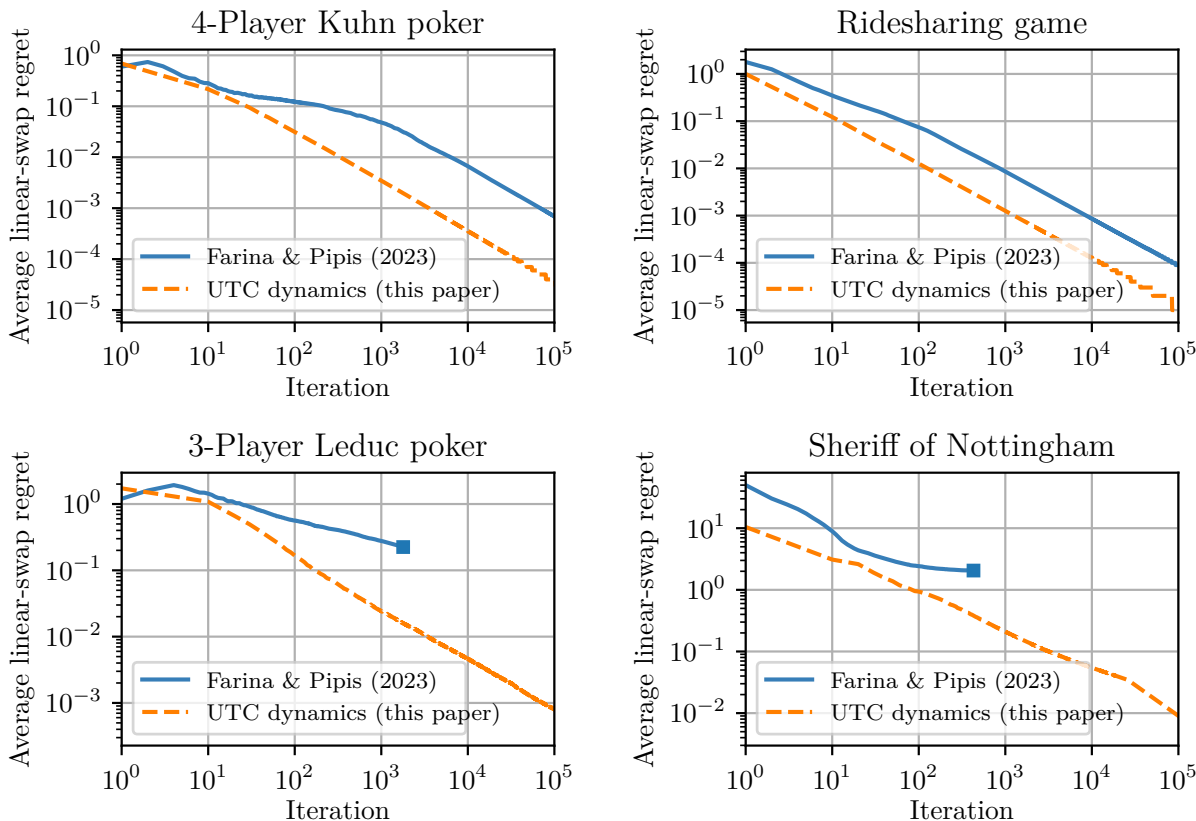
<sup>50</sup>The actions/queries  $\blacktriangle$  makes at **A** and **B** are irrelevant, because  $\blacktriangle$  only cares about maximizing utility, and it always gets utility 0 regardless of what it does. In the depiction of this deviation in Figure 23, the deviator always plays action 1 at **A** and **B**.



**Figure 22:** An example extensive-form game in which communication deviations are a strict subset of UTC deviations. There are two players, P1 ( $\blacktriangle$ ) and P2 ( $\blacktriangledown$ ). Infosets for both players are labeled with capital letters (e.g., **A**) and joined by dotted lines. Actions are labeled with lowercase letters and subscripts (e.g.,  $a_1$ ). P1's utility is labeled on each terminal node. P2's utility is zero everywhere (not labeled). Boxes are chance nodes, at which chance plays uniformly at random.



**Figure 23:** A part of the UTC decision problem for  $\blacktriangle$  corresponding to the same game. Nodes labeled  $\blacktriangle$  are decision points for  $\blacktriangle$ ; boxes are observation points. “...” denotes that the part of the decision problem following that edge has been omitted. Terminal nodes are unmarked. Red edge labels indicate interactions with the mediator; blue edge labels indicate interactions with the game. The profitable untimed deviation discussed in Section 10.4 is indicated by the thick lines. The first action taken in that profitable deviation,  $c_2$ , is not legal for a timed deviator, because a timed deviator must query the mediator once before taking its first action. The matrices (lower-left corner) are the pair of matrices (**A**, **B**) corresponding to that same deviation. All blank entries are 0.



**Figure 24:** Experimental comparison between our dynamics and those of *Farina and Pipis (2023)* for approximating a linear correlated equilibrium in extensive-form games. Each algorithm was run for a maximum of 100,000 iterations or 6 hours, whichever was hit first. Runs that were terminated due to the time limit are marked with a square ■.

## 10.6 Untimed Communication Equilibria

The UTC deviations, like all sets of deviations, give rise to a notion of equilibrium. We define:

**Definition 10.8.** In an extensive-form game, an *untimed private communication equilibrium* is a correlated profile that is a  $(\Phi_i)$ -equilibrium where  $\Phi_i$  is player  $i$ 's set of UTC deviations.

We add the word “private” here in the name to emphasize the fact that the mediator must have a separate interaction with each player—that is, the mediator cannot use its interactions with one player to inform how it gives recommendations to another player. This is enforced by the fact that the equilibrium is a correlated profile. See Part II regarding why this distinction is important.

Defining untimed communication equilibrium without such a privacy restriction seems to be a subtle task, and is orthogonal to and beyond the scope of the present work. However, we will make a few informal comments here. Untimed communication equilibria (without the privacy constraint) are difficult to define in a way that does not quickly collapse to the regular notion of communication equilibrium. In games with three or more players, the mediator is always guaranteed that two of the players have not deviated, and those two players will have messages synchronized with the game clock. Therefore, under reasonable assumptions on how often each player makes moves, the mediator will immediately know if the deviating player is sending out-of-order messages, and this concept would reduce immediately to the regular communication equilibrium. It is entirely unclear how to define a notion of untimed (non-private) communication equilibrium that does not exhibit such a collapse.

In two-player games, it is possible that there is a reasonable way to define untimed communication equilibria.

The above collapse does not apply, because the mediator will not know which player is the one sending out-of-timing messages. However, this definition would still be rather subtle—for example, when do the out-of-order messages arrive to the mediator, relative to the *other player’s* messages? We leave these issues to future work.

## 10.7 Experimental Evaluation

We empirically investigate the performance of our learning dynamics for linear correlated equilibrium, compared to the recent algorithm by [Farina and Pipis \(2023\)](#). We test on four benchmark games:

- 4-player Kuhn poker, a multiplayer variant of the classic benchmark game introduced by [Kuhn \(1950b\)](#). The deck has 5 cards. This game has 3,960 terminal states.
- A ridesharing game, a two-player general-sum game introduced as a benchmark for welfare-maximizing equilibria by [Zhang et al. \(2022b\)](#). This game has 484 terminal states.
- 3-player Leduc poker, a three-player variant of the classic Leduc poker introduced by [Southey et al. \(2005\)](#). Only one bet per round is allowed, and the deck has 6 cards (3 ranks, 2 suits). The game has 4,500 terminal states.
- Sheriff of Nottingham, a two-player general-sum game introduced by [Farina et al. \(2019b\)](#) for its richness of equilibrium points. The smuggler has 10 items, a maximum bribe of 2, and 2 rounds to bargain. The game has 2,376 terminal states.

We run our algorithm based on the UTC polytope, and that of [Farina and Pipis \(2023\)](#) (with the learning rate  $\eta = 0.1$  as used by the authors), for a limit of 100,000 iterations or 6 hours, whichever is hit first. Instead of solving linear programs to find the fixed points, we use power iteration, which is faster in practice. All experiments were run on the same machine with 32GB of RAM and a processor running at a nominal speed of 2.4GHz. For our learning dynamics, we employed the CFR algorithm instantiated with the regret matching<sup>+</sup> ([Tammelin, 2014](#)) regret minimizer at each decision point (see [Theorem 10.6](#)). Experimental results are shown in [Figure 24](#).

One of the most appealing features of our algorithm is that allows CFR-based methods to apply. CFR-based methods are the fastest regret minimizers in practice, so it is unsurprising that using them results in better convergence as seen in [Figure 24](#). Another appealing feature is that our method sidesteps the need of projecting onto the set of transformations. This is in contrast with the algorithm of [Farina and Pipis \(2023\)](#), which requires an expensive projection at every iteration. We observe that this difference results in a dramatic reduction in iteration runtime between the two algorithms, which we quantify in [Table 25](#). So, we remark that when accounting for *time* instead of iterations on the x-axis of the plots in [Figure 24](#), the difference in performance between the algorithms appears even stronger. Such a plot is available in the appendix of the full paper ([Zhang et al., 2024d](#)).

## 10.8 Conclusion

In this paper, we have introduced a new representation for the set of linear deviations when the strategy space is sequence form. Our representation connects linear deviations to the mediator-based framework that is more typically used for correlation concepts in extensive-form games, and therefore gives a reasonable game-theoretic interpretation of what linear equilibria represent. It also leads to state-of-the-art no-linear-regret algorithms, both in theory and in practice.



Game	Our algorithm	Farina and Pipis (2023)	Speedup
4-Player Kuhn poker	5.65ms $\pm$ 0.30ms	195ms $\pm$ 7ms	35 $\times$
Ridesharing game	676 $\mu$ s $\pm$ 80 $\mu$ s	160ms $\pm$ 7ms	237 $\times$
3-Player Leduc poker	42.0ms $\pm$ 0.7ms	12.1s $\pm$ 1.0s	287 $\times$
Sheriff of Nottingham	114ms $\pm$ 16ms	50.2s $\pm$ 9.6s	442 $\times$

**Table 25:** Comparison of average time per iteration. For each combination of game instance and algorithm, the mean and standard deviation of the iteration runtime are noted.

Game	Target gap	Our algorithm	Farina and Pipis (2023)	Speedup
4-Player Kuhn poker	$7 \times 10^{-4}$	32.8s	5h 25m	595 $\times$
Ridesharing game	$9 \times 10^{-5}$	8.89s	4h 07m	1667 $\times$
3-Player Leduc poker	0.224	2.12s	6h 00m	10179 $\times$
Sheriff of Nottingham	2.06	2.00s	6h 00m	10800 $\times$

**Table 26:** Comparison of time taken to achieve a particular linear swap equilibrium gap. The gap is whatever gap was achieved by the algorithm of Farina and Pipis (2023) before termination.

## 11 Efficient $\Phi$ -Regret Minimization with Low-Degree Swap Deviations

### 11.1 Introduction

The long-standing absence of efficient algorithms for computing an NFCE shifted the focus to natural relaxations thereof, which can be understood through the notion of  $\Phi$ -regret (Greenwald and Hall, 2003; Stoltz and Lugosi, 2007; Rakhlin et al., 2011). In particular,  $\Phi$  represents a set of strategy deviations; the richer the set of deviations, the stronger the induced solution concept. When  $\Phi$  contains all possible transformations, one recovers the notion of NFCE—corresponding to *swap regret*, while at the other end of the spectrum, *coarse correlated equilibria* correspond to  $\Phi$  consisting solely of constant transformations (aka. *external regret*). Perhaps the most notable relaxation is the *extensive-form correlated equilibrium (EFCE)* (von Stengel and Forges, 2008), which can be computed exactly in time polynomial in the representation of the game tree (Huang and von Stengel, 2008). Considerable interest in the literature has recently been on *learning dynamics* minimizing  $\Phi$ -regret (e.g., Morrill et al. (2021b,a); Bai et al. (2022); Bernasconi et al. (2023); Noarov et al. (2023); Dudík and Gordon (2009); Gordon et al. (2008); Fujii (2023); Dann et al. (2023); Mansour et al. (2022a); Farina and Pipis (2023)). A key reference point in this line of work is the recent construction of Farina and Pipis (2023), an efficient algorithm minimizing *linear swap regret*—that is, the notion of  $\Phi$ -regret where  $\Phi$  contains all *linear* deviations. Such algorithms lead to an  $\epsilon$ -equilibrium in time polynomial in the game’s description and  $1/\epsilon$ —aka. a fully polynomial-time approximation scheme (FPTAS).

Yet, virtually nothing was known beyond those special cases until recent breakthrough results by Dagan et al. (2024) and Peng and Rubinstein (2024), who introduced a new approach for reducing swap to external regret; unlike earlier reductions (Gordon et al., 2008; Blum and Mansour, 2007; Stoltz and Lugosi, 2005), their algorithm can be implemented efficiently even in certain settings with an exponential number of pure strategies. For extensive-form games, their reduction implies a polynomial-time approximation scheme (PTAS) for computing an  $\epsilon$ -correlated equilibrium; their algorithm has complexity  $N^{\tilde{O}(1/\epsilon)}$  for games of size  $N$ , which is polynomial only when  $\epsilon$  is an absolute constant. Instead, we focus here on algorithms with better complexity  $\text{poly}(N, 1/\epsilon)$ , the typical guarantee one hopes for within the no-regret framework.

## 11.2 Our Results

In this paper, we take an important step toward closing the gap between the aforementioned results by developing parameterized algorithms for minimizing  $\Phi$ -regret. For the sake of exposition, we shall first describe our results for the special case of Bayesian games with two actions per player, and we then treat general extensive-form games.

In this context, each player’s strategy space is a hypercube  $\{0, 1\}^N$ . We introduce the set of *depth- $k$  decision tree deviations*  $\Phi_{\text{DT}}^k$ , which can be described as follows. For each of  $k \in \mathbb{N}$  rounds, the deviator first elects a decision point and receives a recommendation, whereupon the deviator gets to decide which action to follow in that decision point. The set of deviations  $\phi : \{0, 1\}^N \rightarrow [0, 1]^N$  that can be expressed in the above manner is precisely the set of functions representable as (randomized) depth- $k$  decision trees on  $N$  variables. To connect  $\Phi_{\text{DT}}^k$  with the concepts referred to earlier, we clarify that  $k = 1$  corresponds to linear-swap deviations, while  $k = N$  captures all possible swap deviations. Our first result is a parameterized online algorithm minimizing regret with respect to deviations in  $\Phi_{\text{DT}}^k$ . (All our results are in the full feedback model under a strongly adaptive adversary.)

**Theorem 11.1.** *There is an online algorithm incurring (average)  $\Phi_{\text{DT}}^k$ -regret at most  $\epsilon$  in  $N^{O(k)}/\epsilon^2$  rounds with a per-round running time of  $N^{O(k)}/\epsilon$ .*

Next, we consider the set  $\Phi_{\text{poly}}^k$  consisting of all *degree- $k$  polynomials*  $\phi : \{0, 1\}^N \rightarrow \{0, 1\}^N$ . Our result for this class of deviations mirrors the one for  $\Phi_{\text{DT}}^k$ , but with a worse dependence on  $k$ .

**Theorem 11.2.** *There is an online algorithm incurring  $\Phi_{\text{poly}}^k$ -regret at most  $\epsilon$  in  $N^{O(k^3)}/\epsilon^2$  rounds with a per-round running time of  $N^{O(k^3)}/\epsilon$ .*

We find those results surprising; we originally surmised that even for quadratic polynomials ( $k = 2$ ) the underlying online problem would be hard in the regime  $\epsilon = 1/\text{poly}(N)$ . A salient aspect of the above results is that the learner is allowed to output a *probability distribution* over  $\{0, 1\}^N$ . In stark contrast, and perhaps surprisingly, when the learner is constrained to output *behavioral* strategies, that is to say, points in  $[0, 1]^N$ , we show that the problem becomes PPA-hard even for a degree  $k = 2$  (Theorem 11.5). We are not aware of any such hardness results pertaining to a natural online learning problem.

We next expand our scope to arbitrary extensive-form games. We will assume here that the branching factor  $b$  of the game is 2—any game can be transformed as such by incurring a  $\log b$  factor overhead in the depth  $d$  of the game tree. Generalizing  $\Phi_{\text{DT}}^k$  described above, we introduce the set of  *$k$ -mediator deviations*  $\Phi_{\text{med}}^k$ ; informally, the player here has access to  $k$  distinct mediators, which the player can query at any time. Once again, the case  $k = 1$  corresponds to linear-swap deviations. Further, if  $\mathcal{X}$  denotes the set of pure strategies, we let  $\Phi_{\text{poly}}^k$  denote the set of all degree- $k$  deviations  $\mathcal{X} \rightarrow \mathcal{X}$ . We establish similar parameterized results in extensive-form games, but which may now also depend on the depth of the game tree  $d$ .

**Theorem 11.3.** *There is an online algorithm incurring at most an  $\epsilon \Phi_{\text{poly}}^k$  regret in  $N^{O(kd)^3}/\epsilon^2$  rounds with a per-round running time of  $N^{O(kd)^3}/\epsilon$ . For  $\Phi_{\text{med}}^k$  both bounds instead scale as  $N^{O(k)}$ .*

$N$  here again denotes the dimension of the strategy space. For a fixed degree  $k$  and assuming that the game tree is *balanced*, in the sense that  $d = \text{polylog} N$ , the theorem above guarantees a quasipolynomial complexity with respect to  $\Phi_{\text{poly}}^k$ , even when  $\epsilon$  is itself inversely quasipolynomial. The complexity we obtain for  $\Phi_{\text{med}}^k$  is more favorable, being polynomial for any extensive-form game. Finally, in light of the connection between no-regret learning and convergence to correlated equilibria, our results imply parameterized tractability of the equilibrium concepts induced by  $\Phi_{\text{med}}^k$  or  $\Phi_{\text{poly}}^k$ .

## 11.3 Technical Overview

Our starting point is the familiar template of [Gordon et al. \(2008\)](#) for minimizing  $\Phi$ -regret, which consists of two key components. Accordingly, we split our technical overview into two parts.

**Approximate fixed points.** The first key ingredient one requires in the framework of [Gordon et al. \(2008\)](#) is an algorithm for computing an approximate *fixed point* of any function within the set of deviations. In particular, if  $\mathcal{X}$  is the set of pure strategies and  $\text{co } \mathcal{X}$  is the convex hull of  $\mathcal{X}$ , we now work with functions  $\Phi^\delta \ni \phi^\delta : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , so that fixed points exist by virtue of Brouwer’s theorem.<sup>51</sup> This fixed point computation is—at least in some sense—inherent: [Hazan and Kale \(2007\)](#) observed that minimizing  $\Phi^\delta$ -regret is computationally equivalent to computing approximate fixed points of transformations in  $\Phi^\delta$ . Specifically, an efficient algorithm minimizing  $\Phi^\delta$ -regret—with respect to any sequence of utilities—can be used to compute an approximate fixed point of any transformation in  $\Phi^\delta$ . Given that functions in  $\Phi^\delta$  are generally nonlinear, this brings us close to PPAD-hard territory. Indeed, although functions in  $\Phi^\delta$  have a particular structure not directly compatible with prior reductions, we show that they can still simulate generalized circuits even under low-degree deviations. At first glance, this would seem to contradict the recent positive results of [Dagan et al. \(2024\)](#) and [Peng and Rubinstein \(2024\)](#).

It turns out that there is a delicate precondition on the reduction of [Hazan and Kale \(2007\)](#) that makes all the difference: computing approximate fixed points is only necessary if the learner outputs points on  $\text{co } \mathcal{X}$ . In stark contrast, a crucial observation that drives our approach is that a learner who selects a probability distribution over  $\mathcal{X}$  does *not* have to compute (approximate) fixed points of functions in  $\Phi$ . Instead, we show that it is enough to determine what we refer to as an approximate fixed point *in expectation*. More precisely, for a deviation  $\Phi \ni \phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  with an efficient representation, it is enough to compute a distribution  $\pi \in \Delta(\mathcal{X})$  such that  $\mathbb{E}_{\mathbf{x} \sim \pi} \phi(\mathbf{x}) \approx \mathbb{E}_{\mathbf{x} \sim \pi} \mathbf{x}$ . It is quite easy to compute an approximate fixed point in expectation: take any  $\mathbf{x}_1 \in \text{co } \mathcal{X}$ , and consider the sequence  $\mathbf{x}_1, \dots, \mathbf{x}_L \in \text{co } \mathcal{X}$  such that  $\mathbf{x}_{\ell+1} := \mathbb{E}_{\mathbf{x}'_\ell \sim \delta(\mathbf{x}_\ell)} \phi(\mathbf{x}'_\ell)$  for all  $\ell$ , where  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  is a mapping such that  $\mathbb{E}_{\mathbf{x}' \sim \delta(\mathbf{x})} [\mathbf{x}'] = \mathbf{x}$ .<sup>52</sup> Then, for  $\pi := \mathbb{E}_{\ell \in [L]} [\delta(\mathbf{x}_\ell)]$ , we have

$$\mathbb{E}_{\mathbf{x} \sim \pi} [\phi(\mathbf{x}) - \mathbf{x}] = \frac{1}{L} \sum_{\ell=1}^L \mathbb{E}_{\mathbf{x}'_\ell \sim \delta(\mathbf{x}_\ell)} [\phi(\mathbf{x}'_\ell) - \mathbf{x}'_\ell] = \frac{1}{L} \mathbb{E}_{\mathbf{x}'_L \sim \delta(\mathbf{x}_L)} [\phi(\mathbf{x}'_L) - \mathbf{x}_1] = O\left(\frac{1}{L}\right).$$

This procedure can replace the fixed point oracle required by the template of [Gordon et al. \(2008\)](#), which is prohibitive when  $\Phi$  contains nonlinear functions. In fact, even in normal-form games where considering linear deviations suffices, computing a fixed point is relatively expensive, amounting to solving a linear system, dominating the per-iteration complexity. Leveraging instead our new reduction, we obtain the fastest algorithm for computing an approximate correlated equilibrium in the moderate-precision regime (Corollary 11.13). Beyond normal-form games, our observation can be used to speed up many of the prior  $\Phi$ -regret minimizers, which rely on some fixed point operation.

It is worth noting that the discrepancy that has arisen between operating over  $\Delta(\mathcal{X})$  versus  $\text{co } \mathcal{X}$  is quite singular when it comes to regret minimization in extensive-form games. Kuhn’s theorem ([Kuhn, 1953](#)) is often invoked to argue about their equivalence, but in our setting it is the nonlinear nature of deviations in  $\Phi$  that invalidates that equivalence.<sup>53</sup> To tie up the loose ends, we adapt the reduction of [Hazan and Kale \(2007\)](#) to show that minimizing  $\Phi$ -regret over  $\Delta(\mathcal{X})$  necessitates computing approximate fixed points in expectation (Proposition 11.8), and we observe that the reductions of [Dagan et al. \(2024\)](#) and [Peng and Rubinstein \(2024\)](#) are indeed compatible with computing approximate fixed points in expectation (Section 11.8.3).

<sup>51</sup>Here,  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  is used to extend a map  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  to a map  $\phi^\delta : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ .

<sup>52</sup>For technical reasons, it is more convenient to work with functions with domain  $\mathcal{X}$ , in which case we apply  $\delta$  to sample a point in  $\mathcal{X}$  before applying  $\phi$ . For our applications, there is a particularly natural choice for  $\delta$ , as introduced in Definition 11.10.

<sup>53</sup>Kuhn’s theorem is also invalidated in extensive-form games with *imperfect recall* ([Piccione and Rubinstein, 1997](#); [Tewolde et al., 2023](#); [Lambert et al., 2019](#)), in which there is also a genuine difference between mixed and behavioral strategies. In such settings, it is NP-hard to even minimize external regret.

**Regret minimization over the set  $\Phi$ .** The second ingredient prescribed by [Gordon et al. \(2008\)](#) is an algorithm minimizing *external regret* but with respect to the *set of deviations*  $\Phi$ . The crux in this second step lies in the fact that, even in normal-form games,  $\Phi$  contains at least an exponential number of deviations, so black-box reductions are of little use here. Instead, the problem boils down to appropriately leveraging the combinatorial structure of  $\Phi$ , as we explain below.

We will first describe our approach when  $\mathcal{X} = \{0, 1\}^N$ , and we then proceed with the more technical generalization to extensive-form games. The key observation is that regret minimization over  $\Phi_{\text{DT}}^k$  can be viewed as a tree-form decision problem of size  $N^{O(k)}$ . This enables us to rely on usual techniques for dealing with such problems (e.g., ([Zinkevich et al., 2007](#))), leading to a complexity bound of  $N^{O(k)}$ . For the set of low-degree polynomials  $\Phi_{\text{poly}}^k$ , we leverage a result in Boolean analysis relating (randomized) low-depth decision trees with low-degree polynomials ([Theorem 11.16](#)), which implies that  $\Phi_{\text{poly}}^k \subseteq \Phi_{\text{DT}}^{2k^3}$ . Consequently, low-degree polynomials can be reduced to low-depth decision trees, albeit with an overhead in the exponent.

Turning to extensive-form games, we follow a similar blueprint, although there are now additional technical challenges. First, for the set of *k-mediator deviations*  $\Phi_{\text{med}}^k$ , we show that there is a reduction to a particular type of *DAG-form* decision problem of size  $N^{O(k)}$ , a class of problems recently treated by [Zhang et al. \(2023b\)](#). That formulation is more suitable than tree-form decision problems when the number of possible histories far exceeds the number of states, which is precisely the case when the player is gradually querying multiple mediators as the game progresses.

Finally, we establish a reduction from low-degree polynomials to few mediators; namely, we show that  $\Phi_{\text{poly}}^k \subseteq \Phi_{\text{med}}^{O(kd)^3}$ , where we recall that  $d$  is the depth of the game tree. Our basic strategy is to again leverage the connection between low-depth decision trees and low-degree polynomials we described earlier. To do so, we need to cast our problem in terms of functions  $\{0, 1\}^N \rightarrow \{0, 1\}^N$  instead of  $\mathcal{X} \rightarrow \mathcal{X}$ . To that end, we first show how to *extend* a degree- $k$  function  $f : \mathcal{X} \rightarrow \{0, 1\}$  to a degree- $kd$  function  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$ ; that is,  $\bar{f}$  coincides with  $f$  on all points in  $\mathcal{X} \subseteq \{0, 1\}^N$ . This step is where the overhead factor  $d$  comes from. The final technical piece is to show that if each component of  $\phi : \mathcal{X} \rightarrow \mathcal{X}$  can be expressed using  $K$  mediators, the same holds for  $\bar{\phi}$ ; the naive argument here incurs another factor of  $d$ , but we show that this is in fact not necessary.

## 11.4 Hardness of Minimizing $\Phi$ -Regret in Behavioral Strategies

In this section, we show that if the learner is constrained to output in each round a strategy in  $\text{co}\mathcal{X}$ , then there is no efficient algorithm (under standard complexity assumptions) minimizing  $\Phi^\beta$ -regret ([Theorem 11.5](#)); here,  $\beta : \text{co}\mathcal{X} \rightarrow \Delta(\mathcal{X})$  is the behavioral strategy mapping (introduced in the sequel as [Definition 11.10](#)), the expression of which is not important for the purpose of this section. The key connection is an observation by [Hazan and Kale \(2007\)](#), which reveals that any  $\Phi^\beta$ -regret minimizer is inadvertently able to compute approximate fixed points of any deviations in  $\Phi^\beta$ . We then show that the set of induced deviations, even on the hypercube  $\mathcal{X} = \{0, 1\}^N$ , is rich enough to approximate PPAD-hard fixed-point problems.

In this context, consider a transformation  $\Phi^\beta \ni \phi^\beta : [0, 1]^N \rightarrow [0, 1]^N$  for which we want to compute an approximate fixed point  $\mathbf{x} \in \text{co}\mathcal{X}$ ; that is,  $\|\phi^\beta(\mathbf{x}) - \mathbf{x}\|_2 \leq \epsilon$ , for some precision parameter  $\epsilon > 0$ . (It is convenient in the construction below to measure the fixed-point error with respect to  $\|\cdot\|_2$ .) [Hazan and Kale \(2007\)](#) observed that a  $\Phi^\beta$ -regret minimizer can be readily turned into an algorithm for computing fixed points of any function in  $\Phi^\beta$ , as stated formally below. Before we proceed, we remind that here and throughout we operate under a strongly adaptive adversary, which is quite crucial in the construction of [Hazan and Kale \(2007\)](#).

**Proposition 11.4** ([Hazan and Kale, 2007](#)). *Consider a regret minimizer  $\mathcal{R}$  operating over  $[0, 1]^N$ . If  $\mathcal{R}$  runs in time  $\text{poly}(N, 1/\epsilon)$  and guarantees  $\text{REG}_{\Phi^\beta}^T \leq \epsilon$  for any sequence of utilities, then there is a  $\text{poly}(N, 1/\epsilon)$  algorithm for computing an  $(\epsilon\sqrt{N})$ -fixed point of any  $\phi^\beta \in \Phi^\beta$  with respect to  $\|\cdot\|_2$ , assuming that  $\phi^\beta$  can be evaluated in polynomial time.*

Proposition 11.4 significantly circumscribes the class of problems for which efficient  $\Phi^\beta$ -regret minimization is possible, at least when operating in behavioral strategies. Indeed, computing fixed points is in general a well-known (presumably) intractable problem. In our context, the set  $\Phi^\beta$  does not contain arbitrary (Lipschitz continuous) functions  $[0, 1]^N \rightarrow [0, 1]^N$ , but instead contains multilinear functions from  $[0, 1]^N$  to  $[0, 1]^N$ . To establish PPAD-hardness for our problem, we start with a *generalized circuit*, and we show that all gates can be approximately simulated using exclusively gates involving multilinear operations; we defer the formal argument to the appendix of the full paper (Zhang et al., 2024a). As a result, we arrive at the main hardness result of this section.

**Theorem 11.5.** *If  $\mathcal{R}$  outputs strategies in  $[0, 1]^N$ , it is PPAD-hard to guarantee  $\text{REG}_{\Phi^\beta} \leq \epsilon/\sqrt{N}$ , even with respect to low-degree deviations and an absolute constant  $\epsilon > 0$ .*

We also obtain a stronger hardness result under a stronger complexity assumption put forward by Babichenko et al. (2016), which can be found in the appendix of the full paper (Zhang et al., 2024a). At first glance, it may seem that the above results are at odds with the recent positive results of Dagan et al. (2024) and Peng and Rubinfeld (2024), which seemingly obviate the need to compute approximate fixed points. As we have alluded to, the key restriction that drives Theorem 11.5 lies in constraining the learner to output behavioral strategies. In the coming section, we show that there is an interesting twist which justifies the discrepancy highlighted above.

## 11.5 Circumventing Fixed Points

The previous section, and in particular Theorem 11.5, seems to preclude the ability to minimize  $\Phi$ -regret efficiently when the set of (extended) deviations contains nonlinear functions.<sup>54</sup> In this section, we will show how to circumvent this issue via a relaxed notion of what constitutes a fixed point (Definition 11.6). In the sequel, we will work with deviations  $\phi$  with domain  $\mathcal{X}$  instead of  $\text{co } \mathcal{X}$ .

### 11.5.1 Approximate Expected Fixed Points

The key to our construction is to allow the learner to play *distributions* over  $\mathcal{X}$ , not merely points in  $\text{co } \mathcal{X}$ , and to use a relaxed notion of a fixed point, formally introduced below.

---

**Algorithm ExpectedGordon:**  $\Phi$ -regret minimizer using fixed points in expectation, using an external regret minimizer  $\mathcal{R}_\Phi$  on  $\Phi$

---

```

1: initialize  $z^1 \leftarrow \mathbf{1}, t \leftarrow 0$ 
2: procedure NEXTSTRATEGY():
3:    $\phi^{(t)} \leftarrow \mathcal{R}_\Phi.\text{NEXTSTRATEGY}()$ 
4:    $\pi^{(t)} \leftarrow \epsilon$ -expected fixed point of
5:    $\phi^{(t)}$ 
6:   return  $\pi^{(t)}$ 
7: procedure OBSERVEUTILITY( $\mathbf{u}^t$ ):
8:   set  $u_\Phi^{(t)} : \Phi \ni \phi \mapsto \langle \mathbf{u}^t, \mathbb{E}_{\mathbf{x}^{(t)} \sim \pi^{(t)}} \phi(\mathbf{x}^{(t)}) \rangle$ 
9:
10:  $\mathcal{R}_\Phi.\text{OBSERVEUTILITY}(u_\Phi^{(t)})$ 

```

---

**Definition 11.6.** We say that a distribution  $\pi \in \Delta(\mathcal{X})$  is an  $\epsilon$ -*expected fixed point* of  $\phi \in (\text{co } \mathcal{X})^\mathcal{X}$  if  $\|\mathbb{E}_{\mathbf{x} \sim \pi}[\phi(\mathbf{x}) - \mathbf{x}]\|_{\mathcal{X}} \leq \epsilon$ .

The key now is to replace the fixed point oracle in the framework of Gordon et al. (2008) with an oracle that instead returns an  $\epsilon$ -fixed point in expectation per Definition 11.6. The learner otherwise proceeds as in the algorithm of Gordon et al. (2008) (our overall construction is spelled out in Algorithm ExpectedGordon). It

<sup>54</sup>For linear functions, fixed points can be computed exactly via a linear program.

is easy to show, following the proof of [Gordon et al. \(2008\)](#), that a fixed point in expectation is still sufficient to minimize  $\Phi$ -regret.

**Theorem 11.7** ( $\Phi$ -regret with  $\epsilon$ -expected fixed points). *Suppose that the external regret of  $\mathcal{R}_\Phi$  over  $\Phi$  after  $T$  repetitions is at most  $\text{REG}^T$ . Then, the  $\Phi$ -regret of **ExpectedGordon** can be bounded as  $\text{REG}^T + \epsilon$ .*

Analogously to Proposition 11.4, it turns out that there is a certain equivalence between minimizing  $\Phi$  in  $\Delta(\mathcal{X})$  and computing *expected* fixed points:

**Proposition 11.8.** *Consider a regret minimizer  $\mathcal{R}$  operating over  $\Delta(\mathcal{X})$ . If  $\mathcal{R}$  runs in time  $\text{poly}(N, 1/\epsilon)$  and guarantees  $\text{REG}_\Phi^T \leq \epsilon$  for any sequence of utilities, then there is a  $\text{poly}(N, 1/\epsilon)$  algorithm for computing  $(\epsilon D_\mathcal{X})$ -expected fixed points of  $\phi \in \Phi$ , assuming that we can efficiently compute  $\mathbb{E}_{\mathbf{x}^{(t)} \sim \pi^{(t)}}[\phi(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}]$  at any time  $t$ . Here,  $D_\mathcal{X}$  is the diameter of  $\mathcal{X}$  with respect to  $\|\cdot\|_2$ .*

The proof proceeds similarly to Proposition 11.4, and so we defer it to the appendix of the full paper ([Zhang et al., 2024a](#)). Next, we present a method for computing approximate expected fixed points of functions  $\phi \in \Phi$  without having to solve a PPA-hard problem.

## 11.5.2 Extending Deviation Maps to $\text{co } \mathcal{X}$

First, since we will work both over  $\text{co } \mathcal{X}$  and distributions in  $\Delta(\mathcal{X})$ , we need efficient methods for passing between them. To that end, we introduce the following notion.

**Definition 11.9.** A map  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  is

- *consistent* if  $\mathbb{E}_{\mathbf{x}' \sim \delta(\mathbf{x})} \mathbf{x}' = \mathbf{x}$ , and
- *efficient* if, given some  $\phi \in \Phi$  and  $\mathbf{x} \in \text{co } \mathcal{X}$ , it is easy to compute  $\phi^\delta(\mathbf{x}) := \mathbb{E}_{\mathbf{x}' \sim \delta(\mathbf{x})} \phi(\mathbf{x}')$ .

We will call the map  $\phi^\delta : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$  the *extended map* of  $\phi$ .

One may ask why we use this indirect method of defining  $\phi^\delta$  rather than simply directly using the representation of  $\phi$  (for example, as a polynomial) to extend  $\phi$  to  $\text{co } \mathcal{X}$ . The answer is that, even assuming that  $\phi : \mathcal{X} \rightarrow \mathcal{X}$  is represented as a multilinear polynomial (which is the representation assumed in the majority of this paper), naively extending that polynomial to domain  $\text{co } \mathcal{X}$  will not necessarily result in a function  $\bar{\phi} : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ . For an example, consider the decision problem  $\mathcal{X}$  depicted in Figure 27, and consider the function  $\phi : \mathcal{X} \rightarrow \mathcal{X}$  given by  $\phi(\mathbf{x}) = (x_1 + x_3, x_2 x_4, x_2 x_5, x_2, 0)$ . One can easily check by hand that  $\phi$  is indeed a function  $\mathcal{X} \rightarrow \mathcal{X}$ , but also that, for the strategy  $\mathbf{x} = (1/2, 1/2, 0, 1/2, 0) \in \text{co } \mathcal{X}$ , we have  $\phi(\mathbf{x}) = (1/2, 1/4, 0, 1/2, 0) \notin \text{co } \mathcal{X}$ . Thus, we need a more robust way of extending functions  $\mathcal{X} \rightarrow \text{co } \mathcal{X}$  to functions  $\text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , ideally one that is dependent only the function  $\phi$ , not its representation.

We now give two methods of constructing consistent and efficient maps  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  for tree-form strategy sets  $\mathcal{X}$ . The first is the behavioral strategy map.

**Definition 11.10.** The *behavioral strategy map*  $\beta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  is defined as follows:  $\beta(\mathbf{x})$  is the distribution of pure strategies generated by sampling, at each decision point  $j$  for which  $\mathbf{x}[j] > 0$ , an action  $a$  according to the probabilities  $\mathbf{x}[ja]/\mathbf{x}[j]$ . Formally,

$$\beta(\mathbf{x})[\mathbf{y}] := \prod_{j a: \mathbf{x}[j] > 0, \mathbf{y}[ja] = 1} \frac{\mathbf{x}[ja]}{\mathbf{x}[j]}.$$

It is possible for  $\phi^\beta$  to be not a polynomial even when  $\phi$  is a polynomial, because  $\beta$  is *itself* not a polynomial. It is clear that  $\beta$  is consistent. For efficiency, we show the following claim.

**Proposition 11.11.** *Let  $\beta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  be the behavioral strategy map. Let  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  be expressed as a polynomial of degree at most  $k$ , in particular, as a sum of at most  $O(N^k)$  terms. Then there is an algorithm running in time  $N^{O(k)}$  that, given  $\phi$  and  $\mathbf{x} \in \text{co } \mathcal{X}$ , computes  $\phi^\beta(\mathbf{x})$ .*

*Proof.* To compute  $\mathbb{E}_{\mathbf{x}' \sim \beta(\mathbf{x})} \phi(\mathbf{x}')$ , since  $\phi$  is a polynomial, it suffices to compute  $\mathbb{E}_{\mathbf{x}' \sim \beta(\mathbf{x})} m(\mathbf{x}')$  for multilinear monomials  $m$  of degree at most  $K$ , that is, functions of the form  $m_S(\mathbf{x}) := \prod_{z \in S} \mathbf{x}[z]$  where  $S \subseteq \mathcal{Z}$  has size at most  $k$ . There are two cases. First, there are monomials that are clearly identically zero: in particular, if there are two nodes  $ja, ja' \preceq S$  for  $a \neq a'$ , then  $m_S \equiv 0$  because a player cannot play two different actions at  $j$ . For monomials that are not identically zero, we have

$$\mathbb{E}_{\mathbf{x}' \sim \beta(\mathbf{x})} \prod_{ja \in S} \mathbf{x}[ja] = \prod_{ja \preceq S: \mathbf{x}[j] > 0} \frac{\mathbf{x}[ja]}{\mathbf{x}[j]},$$

which is computable in time  $O(kd)$ . Thus, the overall time complexity is  $O(kdN^k) \leq N^{O(k)}$ .  $\square$

The behavioral strategy map is in some sense the *canonical* strategy map: when one writes a tree-form strategy  $\mathbf{x} \in \text{co } \mathcal{X}$  without further elaboration on what distribution  $\Delta(\mathcal{X})$  it is meant to represent, it is often implicitly or explicitly assumed to mean the behavioral strategy.

The behavioral strategy map has the unfortunate property that it usually outputs distributions of exponentially-large support; indeed, if  $\mathbf{x} \in \text{relint } \text{co } \mathcal{X}$  then  $\beta(\mathbf{x})$  is *full-support*.

The second example we propose, which we call a *Carathéodory map*, always outputs low-support distributions. In particular, for any  $\mathbf{x} \in \text{co } \mathcal{X}$ , Carathéodory's theorem on convex hulls guarantees that  $\mathbf{x}$  is a convex combination of  $N$  pure strategies<sup>55</sup>  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathcal{X}$ . Grötschel et al. (1981, Theorem 3.9) moreover showed that there exists an efficient algorithm for computing the appropriate convex combination. Thus, fixing some efficient algorithm for this computational problem, we define a *Carathéodory map*  $\gamma : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  to be any consistent map that returns a distribution of support at most  $N$ . Given such a mapping, computing  $\phi^\gamma(\mathbf{x})$  is easy: one simply writes  $\mathbf{x} = \sum_i \alpha_i \mathbf{x}_i$  by computing  $\gamma(\mathbf{x})$ , and returns  $\phi^\gamma(\mathbf{x}) = \sum_i \alpha_i \phi(\mathbf{x}_i)$ . This only requires a  $\text{poly}(N)$ -time computation of  $\gamma$ , and  $N$  evaluations of the function  $\phi$ . As before, when  $\phi$  is a degree- $k$  polynomial, the time complexity of computing  $\phi^\gamma$  is bounded by  $N^{O(k)}$ .

### 11.5.3 Efficiently Computing Fixed Points in Expectation

Now let  $\delta : \text{co } \mathcal{X} \rightarrow \Delta(\mathcal{X})$  be consistent and efficient. Consider the following algorithm. Given  $\phi \in \Phi$ , select  $\mathbf{x}_1 \in \text{co } \mathcal{X}$  arbitrarily, and then for each  $\ell > 1$  set  $\mathbf{x}_\ell := \phi^\delta(\mathbf{x}_{\ell-1})$ . Finally, select  $\pi := \mathbb{E}_{\ell \sim [L]} \delta(\mathbf{x}_\ell) \in \Delta(\mathcal{X})$  as the output distribution. By a telescopic cancellation, we have

$$\left\| \mathbb{E}_{\mathbf{x} \sim \pi} [\phi(\mathbf{x}) - \mathbf{x}] \right\|_{\mathcal{X}} = \frac{1}{L} \left\| \sum_{\ell=1}^L \mathbb{E}_{\mathbf{x} \sim \delta(\mathbf{x}_\ell)} [\phi(\mathbf{x}) - \mathbf{x}] \right\|_{\mathcal{X}} \leq \frac{1}{L} \left\| \mathbb{E}_{\mathbf{x} \sim \delta(\mathbf{x}_L)} [\phi(\mathbf{x}) - \mathbf{x}_1] \right\|_{\mathcal{X}} \leq \frac{2}{L},$$

as desired. As a result, applying Theorem 11.7, we arrive at the following conclusion.

**Theorem 11.12.** *Let  $\mathcal{R}_\Phi$  be an regret minimizer on  $\Phi$  whose external regret after  $T$  iterations is  $\text{REG}^T$  and whose per-iteration runtime is  $R_1$ , and assume that evaluating the extended map  $\phi^\delta : \text{co } \mathcal{X} \rightarrow \text{co } \mathcal{X}$  takes time  $R_2$ . Then, for every  $\epsilon > 0$ , there is a learning algorithm on  $\mathcal{X}$  whose  $\Phi$ -regret after  $T$  iterations is at most  $\text{REG}^T + \epsilon$  and whose per-iteration runtime is  $O(R_1 + R_2/\epsilon)$ .*

The above result provides a full black-box reduction from  $\Phi$ -regret minimization to external regret minimization on  $\Phi$ , with no need for the possibly-expensive computation of a fixed point. We note that the iterates of the algorithm will depend on the choice of  $\delta$ —for example, setting  $\delta = \beta$  and setting  $\delta = \gamma$  will produce different iterates.

<sup>55</sup>Applying Carathéodory naively would give  $N + 1$  instead of  $N$ , but we can save 1 because the tree-form strategy set is never full-dimensional as a subset of  $\{0, 1\}^N$ .

## 11.5.4 Application for Faster Computation of Correlated Equilibria

An important byproduct of Theorem 11.12 is that it leads to faster algorithms for computing equilibria even in settings where fixed points can be computed in polynomial time. In particular, let us focus for simplicity on  $n$ -player normal-form games with a succinct representation. Here, each player  $i \in [n]$  selects as strategy a probability distribution  $\pi_i \in \Delta(\mathcal{A}_i)$ , where we recall that  $\mathcal{A}_i$  is a finite set of available actions. The expected utility of player  $i$  is given by  $u_i(\pi_1, \dots, \pi_n) := \mathbb{E}_{a_1 \sim \pi_1, \dots, a_n \sim \pi_n} [u_i(a_1, \dots, a_n)]$ , where  $u_i : \mathcal{A}_1 \times \dots \times \mathcal{A}_n \rightarrow [-1, 1]$ . We assume that there is an expectation oracle that computes the vector

$$(u_i(a_i, \pi_{-i}))_{i \in [n], a_i \in \mathcal{A}_i} \quad (13)$$

in time bounded by  $\text{EO}(n, A)$ , where  $A := \max_i |\mathcal{A}_i|$ ; it is known that  $\text{EO}(n, A) \leq \text{poly}(n, A)$  for most interesting classes of succinct classes of games (Papadimitriou and Roughgarden, 2008).

In this context, one can apply Theorem 11.12 in conjunction with the algorithm of Blum and Mansour (2007), instantiated with multiplicative weights update, to arrive at the following result.

**Corollary 11.13.** *For any  $n$ -player game in normal form, there is an algorithm that computes an  $\epsilon$ -correlated equilibrium and runs in time*

$$O\left(\frac{A \log A}{\epsilon^2} \left(\text{EO}(n, A) + n \frac{A^2}{\epsilon}\right)\right).$$

Assuming that the oracle call to (13) ( $\text{EO}(n, A)$ ) does not dominate the per-iteration running time,<sup>56</sup> Corollary 11.13 gives (to our knowledge) the fastest algorithm for computing  $\epsilon$ -correlated equilibria in the moderate-precision regime  $1/A^{\frac{\omega}{2}-1} \leq \epsilon \leq 1/\log A$ , where  $\omega \approx 2.37$  is the exponent of matrix multiplication (Williams et al., 2024); without fast matrix multiplication, which is widely impractical, the lower bound instead reads  $\epsilon \geq 1/\sqrt{A}$ . We provide a detailed comparison to previous algorithms in the appendix of the full paper (Zhang et al., 2024a). Finally, we stress that similar improvements can be obtained beyond normal-form games using the template of Theorem 11.12.

## 11.6 Low-Degree Regret on the Hypercube

In this section, we let  $\mathcal{X}$  be the hypercube  $\{0, 1\}^N$ . Hypercubes are linear transformations of tree-form decision problems; in particular, for Bayesian games in which each player has exactly two actions, the strategy space of every player is, up to linear transformations, a hypercube. Since our results are particularly clean for the hypercube case, we start with that. For an integer  $k \geq 0$ , we define the set of deviations  $\Phi_{\text{DT}}^k$  as follows:

1. The deviator observes an index  $j_0 \in [N]$ .
2. For  $i = 1, \dots, k$ : The deviator selects an index  $j_i \in [N]$ , and observes  $\mathbf{x}[j_i]$ .
3. The deviator selects  $a_0 \in \{0, 1\}$ .

We call attention to the order of operations. In particular, each query  $j$  is allowed to depend on previous observed  $\mathbf{x}[j]$ s. We will assume (WLOG) that the deviator always chooses  $k$  distinct indices  $j$ .

The above process describes a tree-form decision problem of size  $N^{O(k)}$ . Terminal nodes in this decision problem are identified by the original index  $j_0 \in [N]$ , the queries  $j_1, \dots, j_k \in [N]$ , their replies  $a_1, \dots, a_k \in \{0, 1\}$ , and finally the action  $a_0 \in \{0, 1\}$  that is played. Each tree-form strategy  $\mathbf{q}$  in this decision problem defines a function  $\phi_{\mathbf{q}} : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , which is computed by following the strategy  $\mathbf{q}$  through the decision problem. Formally, we have

$$\phi_{\mathbf{q}}(\mathbf{x})[j_0] = \sum_{j_1, a_1, \dots, j_k, a_k} \mathbf{q}[j_0, j_1, a_1, \dots, j_k, a_k, 1] \prod_{i=1}^k \mathbf{x}[j_i, a_i]$$

<sup>56</sup>This is indeed the case in, for example, polymatrix games.



where  $\mathbf{x}[j_i, a_i] = \mathbf{x}[j_i]$  if  $a_i = 1$ , and  $1 - \mathbf{x}[j_i]$  if  $a_i = 0$ . Hence  $\phi_{\mathbf{q}}$  is a degree- $k$  polynomial in  $\mathbf{x}$ .

We define  $\Phi_{\text{DT}}^k$  as the set of such functions  $\phi_{\mathbf{q}}$ . The ‘‘DT’’ in the name  $\Phi_{\text{DT}}^k$  stands for *decision tree*: the set of functions  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  that can be expressed in the above manner is precisely the set of functions representable as (randomized) depth- $k$  decision trees on  $N$  variables.

For intuition, we mention the following special cases:

- $\Phi_{\text{DT}}^0$  is the set of external deviations.
- $\Phi_{\text{DT}}^1$  is the set of all single-query deviations, which [Fujii \(2023\)](#) showed to be equivalent to the set of all linear deviations when  $\mathcal{X}$  is a hypercube.
- $\Phi_{\text{DT}}^N$  is the set of all swap deviations.

Since  $\mathbf{q} \mapsto \phi_{\mathbf{q}}(\mathbf{x})[i]$  is linear, it follows that  $\mathbf{q} \mapsto \langle \mathbf{u}, \phi_{\mathbf{q}}(\mathbf{x}) \rangle$  is also linear for any given  $\mathbf{u} \in \mathbb{R}^n$ . Therefore, a regret minimizer on  $\Phi_{\text{DT}}^k$  can be constructed starting from any regret minimizer for tree-form decision problems; for example, *counterfactual regret minimization* ([Zinkevich et al., 2007](#)), or any of its modern variants.

**Proposition 11.14.** *There is a  $N^{O(k)}$ -time-per-round regret minimizer on  $\Phi_{\text{DT}}^k$  whose external regret is at most  $\epsilon$  after  $N^{O(k)}/\epsilon^2$  rounds.*

Thus, combining with [Proposition 11.11](#) and [Theorem 11.12](#), we immediately obtain a  $\Phi_{\text{DT}}^k$ -regret minimizer with the following complexity.

**Corollary 11.15.** *There is a  $N^{O(k)}/\epsilon$ -time-per-round regret minimizer on  $\mathcal{X}$  whose  $\Phi_{\text{DT}}^k$ -regret is at most  $\epsilon$  after  $N^{O(k)}/\epsilon^2$  rounds.*

Next, we relate depth- $k$  decision trees to low-degree polynomials. Let  $\Phi_{\text{poly}}^k$  be the set of degree- $k$  polynomials  $\phi : \mathcal{X} \rightarrow \mathcal{X}$ . We appeal to a result from the literature on Boolean analysis.

**Theorem 11.16** ([Midrijanis, 2004](#)). *Every degree- $k$  polynomial  $f : \{0, 1\}^N \rightarrow \{0, 1\}$  can be written as a decision tree of depth at most  $2k^3$ .*

In particular,  $\Phi_{\text{poly}}^k \subseteq \Phi_{\text{DT}}^{2k^3}$ . [Corollary 11.15](#) thus also implies a  $\Phi_{\text{poly}}^k$ -regret minimizer:

**Corollary 11.17.** *Let  $\mathcal{X} = \{0, 1\}^N$ . There is an  $N^{O(k^3)}/\epsilon$ -time-per-round regret minimizer on  $\mathcal{X}$  whose  $\Phi_{\text{poly}}^k$ -regret is at most  $\epsilon$  after  $N^{O(k^3)}/\epsilon^2$  rounds.*

It is reasonable to ask whether the above result generalizes to polynomials  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ . Indeed, when  $k \leq 1$  or  $k = N$ , every degree- $k$  polynomials  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  can be written as a convex combination of degree- $k$  polynomials  $\phi : \mathcal{X} \rightarrow \mathcal{X}$ , even for arbitrary tree-form decision problems.<sup>57</sup> However, this is not generally true. A brute-force search shows that the polynomial  $\phi : \{0, 1\}^4 \rightarrow [0, 1]^4$  given by

$$\phi(x_1, x_2, x_3, x_4) = x_1 - x_1x_2 - \frac{1}{2}x_1x_3 + \frac{1}{2}x_2x_3 + \frac{1}{2}x_3x_4$$

is quadratic, but it is not a convex combination of quadratics whose range is  $\{0, 1\}^4$ . Perhaps more glaringly, if one could efficiently represent the set of quadratic functions  $\phi : \{0, 1\}^N \rightarrow [0, 1]^N$ , then one could in particular *decide* whether a given quadratic function  $\phi : \{0, 1\}^N \rightarrow \mathbb{R}^N$  has range  $[0, 1]^N$ . But this is a coNP-complete problem.

<sup>57</sup>For degree 0 and  $N$  this is trivial; for degree 1 it is due to [Zhang et al. \(2024d\)](#).

## 11.7 Extensive-Form Games

The goal of this section is to extend the results in the previous section to the *extensive-form* setting, that is, to generalize them to all tree-form decision problems.

### 11.7.1 Interleaving Decision Problems

In this section, we define operations of merging decision problems that will be very useful as notation in the subsequent discussion. Given two decision problems  $\mathcal{X}$  and  $\mathcal{Y}$  with node sets  $\mathcal{S}_1$  and  $\mathcal{S}_2$  respectively, we introduce three new decision problems.

**Definition 11.18.** The *dual*  $\bar{\mathcal{X}}$  of  $\mathcal{X}$  is the decision problem identical to  $\mathcal{X}$ , except that the decision points and observation points have been swapped.

**Definition 11.19.** The *interleaving*  $\mathcal{X} \otimes \mathcal{Y}$  is the tree-form decision problem defined as follows. There is a state  $\mathbf{s} = (s_1, s_2) \in \mathcal{S}_1 \times \mathcal{S}_2$ . The root state is the tuple  $(\emptyset, \emptyset)$ . The decision problem is defined by the player being able to interact with *both* decision problems, in the following manner. At each state  $\mathbf{s} = (s_1, s_2)$ :

- If  $s_1$  and  $s_2$  are both terminal then so is  $\mathbf{s}$ . Otherwise:
- If either of the  $s_i$ s is an observation point, then so is  $\mathbf{s}$ . The children are the states  $(s'_i, s_{-i})$  where  $s'_i$  is a child of  $s_i$ . (If both  $s_i$ s are observation points, both children  $s'_1, s'_2$  are selected simultaneously. This can only happen at the root.)
- Otherwise,  $\mathbf{s}$  is a decision point. The player selects an index  $i \in \{1, 2\}$  at which to act, and a child  $s'_i$  to transition to. The next state is  $(s'_i, s_{-i})$ .

It follows immediately from definitions that  $\bar{\bar{\mathcal{X}}} = \mathcal{X}$ , and  $\otimes$  is associative and commutative. The name and notation for the dual is inspired by the observation that  $\langle \mathbf{x}, \mathbf{y} \rangle = 1$  for all  $\mathbf{x} \in \mathcal{X}$  and  $\mathbf{y} \in \bar{\mathcal{X}}$ : indeed, the component-wise product  $\mathbf{x}[z]\mathbf{y}[z]$  is exactly the probability that one reaches terminal node  $z$  by following strategy  $\mathbf{x}$  at  $\mathcal{X}$ 's decision points and  $\mathbf{y}$  at  $\mathcal{X}$ 's observation points. We also define the notation  $\mathcal{X}^{\otimes k} := \mathcal{X} \otimes \dots \otimes \mathcal{X}$ , where there are  $k$  copies of  $\mathcal{X}$ .

In  $\mathcal{X} \otimes \mathcal{Y}$ , the same state  $(s_1, s_2)$  can be reachable through possibly exponentially many paths, because the learner may choose to interleave actions in  $\mathcal{X}$  with actions in  $\mathcal{Y}$  in any order. Thus, each state  $(s_1, s_2)$  corresponds to actually exponentially many histories in  $\mathcal{X} \otimes \mathcal{Y}$ . In the discussion below, we will therefore carefully distinguish between *histories* and *states*.

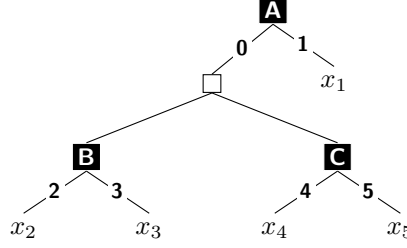
In light of the above exponential gap between histories and states, it seems wasteful to represent  $\mathcal{X} \otimes \mathcal{Y}$  as a tree. Indeed, Zhang et al. (2023b) studied DAG-form decision problems, and showed that regret minimization on them is possible so long as the DAG obeys some natural properties. We state here an immediate consequence of their analysis, which we will use as a black box. Intuitively, the below result states that, as long as utility vectors also only depend on the (terminal) state  $\mathbf{s}$  that is reached, regret minimization on an arbitrary interleaving of decision problems  $\mathcal{X}_1 \otimes \dots \otimes \mathcal{X}_k$  is possible, and the complexity depends only on the number of states.

**Theorem 11.20** (Consequence of Zhang et al., 2023b, Corollary A.4). *Let  $\mathcal{X} := \mathcal{X}_1 \otimes \dots \otimes \mathcal{X}_k$ , where  $\mathcal{X}_i$  has terminal node set  $\mathcal{Z}_i$ . Let  $\mathcal{Z} := \mathcal{Z}_1 \times \dots \times \mathcal{Z}_k$  be the set of terminal states for  $\mathcal{X}$ . For each such terminal state  $z \in \mathcal{Z}$ , let  $V(z)$  be the set of histories of  $\mathcal{X}$  whose state is  $z$ . Define the projection  $\pi : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{Z}}$  by*

$$\pi(\mathbf{x})[z] = \sum_{v \in V(z)} \mathbf{x}[v].$$

*Then there exists an efficient regret minimizer on  $\pi(\mathcal{X}) := \{\pi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\} \subset \mathbb{R}^{\mathcal{Z}}$ : its per-round complexity is  $\text{poly}(|\mathcal{Z}|)$ , and its regret is  $\epsilon$  after  $\text{poly}(|\mathcal{Z}|/\epsilon^2)$  rounds.*

Whenever we speak of regret minimizing on interleavings, it will always be the case that utility vectors depend only on the state, so we will always be able to apply the above result. We will call vectors in  $\pi(\mathcal{X})$  *reduced*



**Figure 27:** An example of a tree-form decision problem. Decision points are black squares with white text labels; observataion points are white squares. Edges are labeled with action names, which are numbers. Pure strategies in this decision problem are identified with vectors  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) \in \{0, 1\}^5$  satisfying  $1 - x_1 = x_2 + x_3 = x_4 + x_5$ .

strategies.

Before proceeding, it is instructive to describe in more detail a result of Zhang et al. (2024d), which we will also use later, in the language of this section. Let  $\mathcal{X}$  and  $\mathcal{Y}$  be any two decision problems with terminal node sets  $\mathcal{Z}_1$  and  $\mathcal{Z}_2$  respectively. A reduced strategy  $\mathbf{q} \in \pi(\mathcal{X} \otimes \bar{\mathcal{Y}})$  induces a linear map  $\phi_{\mathbf{q}} : \mathcal{Y} \rightarrow \text{co } \mathcal{X}$ , given by

$$\phi_{\mathbf{q}}(\mathbf{y})[z_1] = \sum_{z_2 \in \mathcal{Z}_2} \mathbf{q}[z_1, z_2] \mathbf{y}[z_2].$$

It is instructive to think, as Zhang et al. (2024d) detailed extensively in their paper, about what strategies  $\mathbf{q} \in \pi(\mathcal{X} \otimes \bar{\mathcal{Y}})$  represent, and why they induce the linear maps  $\phi_{\mathbf{q}}$ . Decision points  $j$  in  $\mathcal{Y}$  become observation points in  $\mathcal{X} \otimes \bar{\mathcal{Y}}$ —at these observation points, the player should observe the action taken by strategy  $\mathbf{y}$  at  $j$ . The player in  $\mathcal{X} \otimes \bar{\mathcal{Y}}$  is given the ability to *query* the strategy  $\mathbf{y}$  by *taking the role of the environment* in  $\mathcal{Y}$ , while the environment, holding a strategy  $\mathbf{y} \in \mathcal{Y}$ , takes the role of the player and answers decision point queries with the actions that it plays. The player then uses these queries to inform how it plays in the true decision problem  $\mathcal{X}$ . This is the sense in which  $\mathbf{q}$  induces a map  $\phi_{\mathbf{q}}$ : the output  $\phi_{\mathbf{q}}(\mathbf{y})$  is precisely the strategy that would be played if the environment in  $\mathcal{X} \otimes \bar{\mathcal{Y}}$  answers the queries by consulting the strategy  $\mathbf{y}$ . We will call a device that answers queries using strategy  $\mathbf{y}$  a *mediator holding strategy  $\mathbf{y}$* . Zhang et al. (2024d) then showed the following fact, which we will use critically and repeatedly in the rest of this paper.

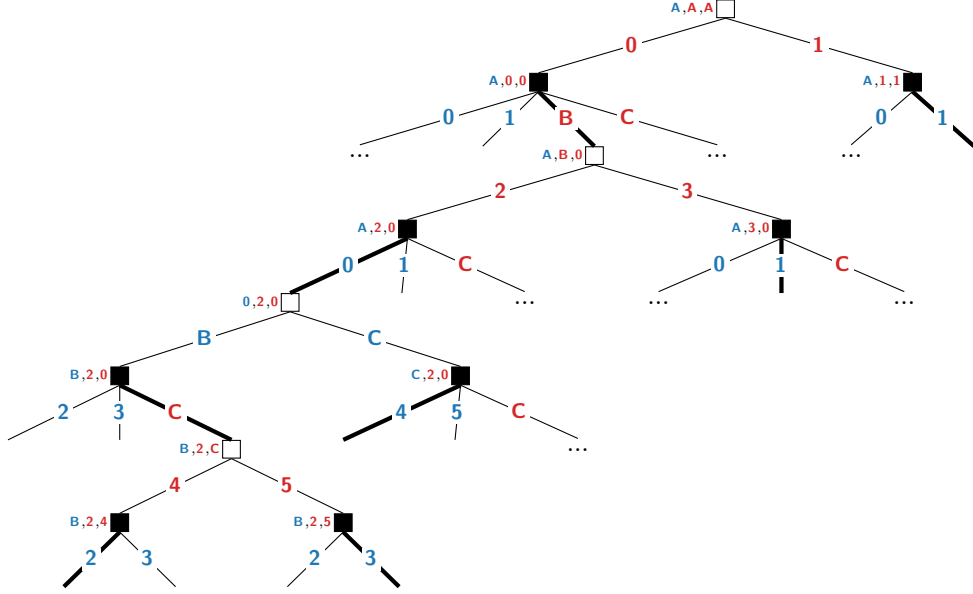
**Theorem 11.21** (Zhang et al., 2024d, Theorem A.2). *Every linear map  $\phi : \mathcal{Y} \rightarrow \text{co } \mathcal{X}$  is induced by some reduced strategy  $\mathbf{q} \in \pi(\mathcal{X} \otimes \bar{\mathcal{Y}})$ .*

## 11.7.2 Efficient Low-Degree Swap-Regret Minimization in Extensive-Form Games

We now proceed with generalizing the results of Section 11.6 to extensive-form games.

Let  $\mathcal{X}$  be any decision problem of dimension  $N$  and depth  $d$ . We will assume WLOG that every decision point in  $\mathcal{X}$  has branching factor exactly 2. This is without loss of generality, but it incurs a loss of  $O(\log b)$ , where  $b$  is the original branching factor, in the depth. Thus, in the below bounds, when  $d$  appears, it should be read as  $O(d \log b)$ .

Using the notation we have now established, we define the set of  $k$ -mediator deviations  $\Phi_{\text{med}}^k$  as the set of reduced strategies in the decision problem  $\mathcal{X} \otimes \bar{\mathcal{X}}^{\otimes k}$ . That is, the player has access to not one but  $k$  mediators, all holding strategy  $\mathbf{x}$ , which the player can query at any time. This is a significant advantage over having just one mediator, since the player can send different queries to each of the  $k$  mediators (who must all reply according to  $\mathbf{x}$ ), and therefore can learn more about the strategy  $\mathbf{x}$  than it could have otherwise. We will call the responses given by the mediator *action recommendations*.



**Figure 28:** A representation of the deviation  $\phi(\mathbf{x}) = (x_1 + x_3, x_2x_4, x_2x_5, x_2, 0)$  (discussed in Section 11.5.2) in the decision problem  $\mathcal{X}$  in Figure 27, as a strategy in  $\mathcal{X} \otimes \bar{\mathcal{X}} \otimes \bar{\mathcal{X}}$ , i.e., with  $k = 2$  mediators. (For an example of a one-mediator deviation, see Zhang et al. (2024d, Figure 1).) Again, black squares are decision nodes and white squares are observation nodes. Nodes are labeled with their state representations: the state in  $\mathcal{X}$  first (in blue), and the two mediator states after (in red). Similarly, blue edge labels indicate interactions with the decision problem (i.e., playing actions and receiving observations in  $\mathcal{X}$ ), and red edge labels indicate interactions with the mediators (i.e., querying and receiving action recommendations from the mediators). Redundant edges (such as those in which the decision problem in  $\mathcal{X}$  has terminated) are omitted. The deviation is shown in thick black lines. For example,  $\phi_2(\mathbf{x}) = x_2x_4$  because the only state in which the deviator plays action 2 is when the mediator state is (2,4).  $\phi_1(\mathbf{x}) = x_1 + x_3$  because the deviator plays action 1 at mediator states (1,1) and (3,0), which would give the formula  $\phi_1(\mathbf{x}) = x_1^2 + x_3x_0$  (where  $x_0 := 1 - x_1$ ), but one can easily check that  $x_1^2 + x_3x_0 = x_1 + x_3$  for all  $\mathbf{x} \in \mathcal{X}$ .

Reduced strategies  $\mathbf{q} \in \pi(\mathcal{X} \otimes \bar{\mathcal{X}}^{\otimes k})$ , once again, induce functions  $\phi_{\mathbf{q}} : \mathcal{X} \rightarrow \text{co } \mathcal{X}$  given by

$$\phi_{\mathbf{q}}(\mathbf{x})[z] = \sum_{z_1, \dots, z_k} \mathbf{q}[z, z_1, \dots, z_k] \prod_{i=1}^k \mathbf{x}[z_i],$$

and again in particular we have that  $\phi_{\mathbf{q}}$  is a degree- $k$  polynomial. We define  $\Phi_{\text{med}}^k$  as the set of such deviations. For intuition, we once again pose a few special cases:

- When the original decision problem's decision space is  $\Delta_2^N$  (i.e., the decision problem consists of a single root observation point with  $N$  children, each of which is a decision point with two actions), we have  $\Phi_{\text{DT}}^k = \Phi_{\text{med}}^k$ . Thus, the results in this section strictly generalize those in the previous section.
- $\Phi_{\text{med}}^0$  and  $\Phi_{\text{med}}^N$  are, as before, the sets of external and swap deviations respectively.
- $\Phi_{\text{med}}^1$  is, by Theorem 11.21, the set of all linear deviations.

In this context, applying Theorem 11.20 gives an efficient  $\Phi_{\text{med}}^k$ -regret minimizer:

**Theorem 11.22.** *There is an  $N^{O(k)}$ -time-per-round regret minimizer on  $\Phi_{\text{med}}^k$  whose external regret is at most  $\epsilon$  after  $N^{O(k)}/\epsilon^2$  rounds.*

Thus, once again Proposition 11.11 and Theorem 11.12 have the following consequence.

**Corollary 11.23.** *There is a  $N^{O(k)}/\epsilon$ -time-per-round regret minimizer on  $\mathcal{X}$  whose  $\Phi_{\text{med}}^k$ -regret is at most  $\epsilon$  after  $N^{O(k)}/\epsilon^2$  rounds.*

Next, we discuss extensions of our result to low-degree polynomials. Unfortunately, we cannot directly apply Theorem 11.16 to conclude the existence of a regret minimizer on  $\mathcal{X}$  with  $\Phi_{\text{poly}}^k$ -regret growing as  $N^{O(k^3)}/\epsilon^2$ . There are two issues in attempting to do so.

First, when  $\mathcal{X}$  is not the hypercube, polynomials  $f : \mathcal{X} \rightarrow \{0, 1\}$  are not total functions. That is, it is not necessarily the case that degree- $k$  polynomials  $f : \mathcal{X} \rightarrow \{0, 1\}$  can be extended to degree- $k$  polynomials  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$ , which is required in order to apply Theorem 11.16.<sup>58</sup> For an example of this, consider  $\mathcal{X} = \mathcal{D}_4$  where  $\mathcal{D}_N$  is the standard basis in  $\mathbb{R}^N$ , that is,  $\mathcal{D}_N = \{e_i : i \in [N]\}$  where  $e_i \in \mathbb{R}^N$  is the  $i$ th basis vector (in other words,  $\mathcal{D}_N$  is the set of vertices of the probability simplex  $\Delta(N)$ ). Let  $f : \mathcal{D}_4 \rightarrow \{0, 1\}$  given by  $f(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$ . Then  $f$  is linear, but there is no linear  $\bar{f} : \{0, 1\}^4 \rightarrow \{0, 1\}$  extending  $f$ . Indeed, there is a more general manifestation of this phenomenon:

**Proposition 11.24.** *For every  $N$ , there exists a linear map  $f : \mathcal{D}_N \rightarrow \{0, 1\}$  such that any extension  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$  of  $f$  must have degree at least  $\Omega(\log N)$ .*

*Proof.* Let  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$  be any degree- $k$  function. By Theorem 3.4 of O’Donnell (2014),  $\bar{f}$  is a  $k2^k$ -junta, that is,  $\bar{f}(\mathbf{x})$  depends on at most  $k2^k$  entries of  $\mathbf{x}$ . Now consider the map  $f : \mathcal{D}_N \rightarrow \{0, 1\}$  given by  $f(\mathbf{x}) = \sum_{i \leq N/2} \mathbf{x}_i$ . Let  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$  be an extension of  $f$ . Then  $\bar{f}$  depends on at least  $N/2 - 1$  inputs: if  $\bar{f}(\mathbf{0}) = 0$  then  $\bar{f}$  depends on at least  $\mathbf{x}_1, \dots, \mathbf{x}_{\lfloor N/2 \rfloor}$ , and if  $\bar{f}(\mathbf{0}) = 1$  then  $\bar{f}$  depends on at least  $\mathbf{x}_{\lfloor N/2 \rfloor + 1}, \dots, \mathbf{x}_N$ . Thus, we have  $N/2 - 1 \leq k2^k$ , which upon rearranging gives  $k \geq \Omega(\log N)$ .  $\square$

The second issue is the following. Suppose that  $K$  mediators were enough to represent a function  $f : \mathcal{X} \rightarrow \{0, 1\}$ . How does one then represent a function  $\phi : \mathcal{X} \rightarrow \mathcal{X}$ ? Each coordinate of  $\phi$  could be represented using  $K$  mediators, but that need not mean the whole function can. In game-theoretic terms, representing a coordinate of  $\phi(\mathbf{x})$  allows the player to play a single action, not necessarily the whole game. Naively, playing the whole game would seem to require  $Kd$  mediators:  $K$  mediators for every level of the decision tree, to compute which action to take at each level.

We will show that it is possible to circumvent both of these issues: the first with a loss of  $O(d)$  in the degree of the polynomial that is representable, and the second with no additional loss. In particular, we state our main result below.

**Theorem 11.25.**  $\Phi_{\text{poly}}^k \subseteq \Phi_{\text{med}}^{O(kd)^3}$ . *Therefore, for every  $k$ , there is a  $N^{O(kd)^3}/\epsilon$ -time-per-round algorithm whose  $\Phi_{\text{poly}}^k$ -regret at most  $\epsilon$  after  $N^{O(kd)^3}/\epsilon$  rounds.*

<sup>58</sup>Formally, we call  $\bar{f} : \{0, 1\}^N \rightarrow \{0, 1\}$  an *extension* of  $f : \mathcal{X} \rightarrow \{0, 1\}$  if  $\bar{f}$  agrees with  $f$  on  $\mathcal{X}$ .

## 11.8 Discussion and Applications

In this section, we discuss various implications and make several remarks about the framework and results that we have introduced.

### 11.8.1 Convergence to Correlated Equilibria

From Theorems 11.22 and 11.25 and Proposition 9.4 it follows that, given a game  $\Gamma$  where the dimension of each player's decision problem is at most  $N$ , we have the following results.

**Corollary 11.26.** *An  $\epsilon$ - $k$ -mediator equilibrium can be computed in time  $N^{O(k)}/\epsilon^3$ .*

**Corollary 11.27.** *An  $\epsilon$ -degree- $k$ -swap equilibrium can be computed in time  $N^{O(kd)^3}/\epsilon^3$ .*

The issue of representing the induced correlated distribution is discussed in the appendix of the full paper (Zhang et al., 2024a).

### 11.8.2 Strict Hierarchy of Equilibrium Concepts

Let  $c \in \{\text{med}, \text{poly}\}$ . For every  $k \geq 0$ , let  $\mathcal{E}_c^k(\Gamma)$  be the set of  $\Phi_c^k$ -equilibria in  $\Gamma$ . It is clear from definitions that  $\mathcal{E}_c^k(\Gamma) \subseteq \mathcal{E}_c^{k-1}(\Gamma)$ . Further, even for normal-form games, it is known that coarse-correlated equilibria are not generally equivalent to correlated equilibria, so at least one of these inclusions is strict in some games. We now show that *all* of these inclusions are strict, so that the deviations  $\Phi_c^k$  form a *strict* hierarchy of equilibria.<sup>59</sup>

**Proposition 11.28.** *For every  $k \geq 1$ , there exists a game  $\Gamma$  such that  $\mathcal{E}_c^k(\Gamma) \subsetneq \mathcal{E}_c^{k-1}(\Gamma)$ .*

*Proof.* Consider the two-player game  $\Gamma$  defined as follows.

- P1's strategy space is  $\mathcal{X} = \{-1, 1\}^k$ . Player 2's strategy space is simply  $\mathcal{Y} = \{-1, 1\}$ .<sup>60</sup>
- P1's utility function is  $u_1(\mathbf{x}, y) = x_1 y$ . That is, P1 would like to set  $x_1 = y$ . P2 gets no utility.

Consider the correlated profile  $\pi$  defined as follows:  $\pi$  is uniform over the  $2^k$  pure profiles  $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$  such that  $y = x_1 x_2 \dots x_k$ . P1's expected utility is clearly 0, and there is a swap (*i.e.*,  $\Phi_c^k$ ) deviation that yields a profit of 1, namely  $\mathbf{x} \mapsto (x_1 x_2 \dots x_k, \dots)$ . (it does not matter what the swap deviation plays at coordinates other than the first one.) But, since all the  $x_i$ s are independent, no function of degree less than  $k$  can have positive correlation with  $x_1 x_2 \dots x_k$ , and thus, there are no profitable deviations of degree less than  $k$ . Thus,  $\pi$  is a  $\Phi_c^{k-1}$ -equilibrium, but not a  $\Phi_c^k$ -equilibrium.  $\square$

<sup>59</sup>The below result constructs a game that depends on  $k$ . It is *not* the case that there exists a single game for which the inclusion hierarchy is strict: for example, for  $k \geq N$ , the set  $\Phi_c^k$  will already contain all the deviations, so  $\mathcal{E}_c^k(\Gamma) = \mathcal{E}_c^N(\Gamma)$  for every  $k \geq N$ .

<sup>60</sup>These strategy spaces are not technically tree-form strategy spaces, but they are linear transformations of tree-form strategy spaces, so one can also rephrase this argument over tree-form strategy spaces. For cleanliness of notation, we stick to  $\{-1, 1\}^k$  as the strategy space.

### 11.8.3 Characterization of Recent Low-Swap-Regret Algorithms in Our Framework

We have, throughout this paper, introduced and used a framework of  $\Phi$ -regret that involves fixed points in expectation. Proposition 11.8 shows that the ability to compute fixed points in expectation is in some sense necessary for the ability to minimize  $\Phi$ -regret. It is instructive to briefly discuss how the recent swap-regret-minimizing algorithm of Dagan et al. (2024) and Peng and Rubinstein (2024) fits into this framework. Their algorithm makes no explicit reference to fixed-point computation, nor to the minimization of external regret over swap deviations  $\phi$ —they do not explicitly invoke the framework we use in this paper, nor that of Gordon et al. (2008). Where is the expected fixed point hidden, then? While we will not present their entire construction here, it suffices to state the following property of it. At every round  $t$ , the learner outputs a distribution  $\pi^{(t)} \in \Delta(\mathcal{X})$  that is uniform on  $L$  strategies  $\mathbf{x}^{(t,1)}, \dots, \mathbf{x}^{(t,L)}$ . The way to map this into our framework is to consider  $\pi^{(t)}$  an approximate fixed point in expectation of the “function”<sup>61</sup>  $\phi^{(t)}$  that maps  $\mathbf{x}^{(t,\ell)} \mapsto \mathbf{x}^{(t,\ell+1)}$  for each  $\ell = 1, \dots, L - 1$ . With this choice of  $\phi^{(t)}$ , their algorithm indeed fits into our framework.

### 11.8.4 Revelation Principles (or Lack Thereof)

Most notions of correlated equilibrium obey some form of *revelation principle*. Informally, one can treat a player attempting to deviate profitably from a correlated equilibrium as an interaction between a mediator (who sends useful information to the player) and the player (who tries to play optimally by using the mediator). When studying the regret of online algorithms, one assumes that the interaction with the mediator is *canonical*: the mediator holds with it some sampled strategy profile  $(\mathbf{x}_1, \dots, \mathbf{x}_n) \sim \pi$ , and in equilibrium every player indeed plays  $\mathbf{x}_i$ . We say that the *revelation principle holds* for a particular notion of equilibrium if allowing *non-canonical* equilibria would not expand the set of equilibria. In the appendix of the full paper (Zhang et al., 2024a), we give a rather general formalization of this notion, which is enough to encompass all the notions of correlated equilibrium discussed in the paper. We show that, in this formalism, the revelation principle *does not* hold for  $k$ -mediator equilibria or degree- $k$  swap equilibria when  $k > 1$ , and indeed in both cases the set of outcomes that can be induced by non-canonical equilibria is the set of *linear-swap* outcomes.

## 11.9 Conclusions and Future Work

In conclusion, we have provided a new family of parameterized algorithms for minimizing  $\Phi$ -regret in extensive-form games. Our results capture perhaps the most natural class of functions interpolating between linear-swap and swap deviations, namely degree- $k$  deviations. Along the way, we refined the usual template for minimizing  $\Phi$ -regret—taught in many courses on algorithmic game theory and online learning—which revolves around (approximate) fixed points (Gordon et al., 2008; Blum and Mansour, 2007; Stoltz and Lugosi, 2005). Instead, we showed that it suffices to rely on a relaxation that we refer to as an approximate fixed point in expectation, which—unlike actual fixed points—can always be computed efficiently. Our refinement of the usual template for minimizing  $\Phi$ -regret has an independent interest beyond extensive-form games. For example, it can speed up the computation of approximate correlated equilibria even in normal-form games, as it obviates the need to solve a linear system in every round. As in the recent works by Dagan et al. (2024) and Peng and Rubinstein (2024), a crucial feature of our approach is to allow the learner to select a distribution over pure strategies, for otherwise we showed that regret minimization immediately becomes PPAD-hard (under a strongly adaptive adversary).

There are many interesting avenues for future research. First, the complexity of our algorithm pertaining degree- $k$  deviations depends exponentially on the depth of the game tree. We suspect that such a dependency could be superfluous, *i.e.*, that there should be an  $N^{\text{poly}(k)}$ -round algorithm for minimizing regret against degree- $k$  deviations.

It would also be interesting to devise parameterized algorithms for degree- $k$  deviations that recover as a special case the PTAS of Peng and Rubinstein (2024) and Dagan et al. (2024), so as to smoothly interpolate

---

<sup>61</sup>“Function” is in quotes because the stated  $\phi$  may not be a function at all; for example, the sequence  $\mathbf{x}^{(t,1)}, \dots, \mathbf{x}^{(t,L)}$  may contain repeats yet be aperiodic.

between existing results for linear-swap regret (Farina and Pipis, 2023) and the aforementioned results for swap regret.

Finally, perhaps the most important question is to understand the computational complexity of computing  $\Phi$ -equilibria in extensive-form games. In particular, our results raise the interesting question of whether there is an algorithm (in the centralized model) for computing in polynomial time an *exact* correlated equilibrium induced by low-degree deviations. Extending the paradigm of Papadimitriou and Roughgarden (2008) in that setting presents several challenges, not least because computing fixed points—which are crucial for implementing the separation oracle (Papadimitriou and Roughgarden, 2008)—is now computationally hard. Relatedly, we suspect that there is an inherent connection between fixed points and correlated equilibria, in the spirit of the equivalence established by Hazan and Kale (2007) in the adversarial regime.

## 12 Steering No-Regret Learners to a Desired Equilibrium

### 12.1 Introduction

Any student of game theory learns that games can have multiple equilibria of different quality—for example, in terms of social welfare. As such, a foundational problem that has received tremendous interest in the literature revolves around characterizing the quality of the equilibrium reached under *no-regret* learning dynamics. The outlook that has emerged from this endeavor, however, is discouraging: typical learning algorithms can fail spectacularly at reaching desirable equilibria. This is rather dramatically illustrated in the example of Figure 29 (second panel). Learning agents initialized at either A, B, or C will in fact converge to the *Pareto-pessimal* Nash equilibrium of the game (bottom-left corner); only an initialization close to the Pareto-dominant equilibrium (such as D in the top-right corner) will end up with the desired outcome.

Our goal in this paper is to develop methods to *steer* learning agents toward better equilibrium outcomes. To do so, we will use a *mediator* that can observe the agents playing the game, give *advice* to the agents (in the form of action recommendations), and *pay* the agents as a function of what actions they played. Our goal is to develop algorithms that allow the mediator to steer agents to a target equilibrium, while not spending too much money doing so. Critically, our only assumption on the agents’ behavior is that they have no regret in hindsight. This is a fairly mild assumption compared to the assumptions made by many past papers on similar topics. We will elaborate on the comparison to related work in the full paper (Zhang et al., 2024b).

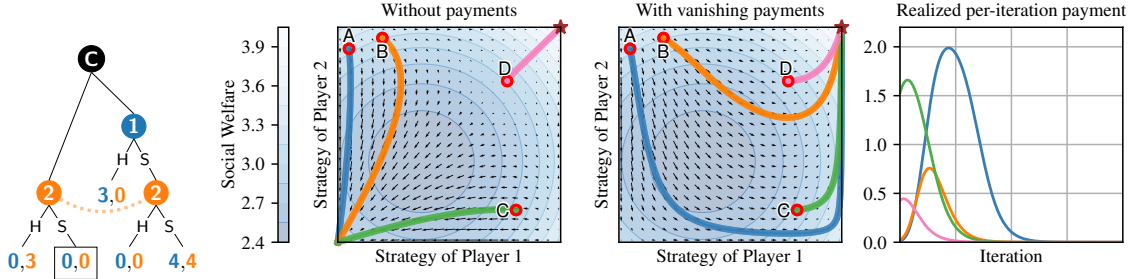
Beyond the obvious relation to equilibrium selection, our model also has implications for the problem of *information design* and Bayesian persuasion (e.g., Kamenica and Gentzkow 2011). Indeed, we will show that we can steer players not only to any Nash equilibrium but to any *Bayes-correlated equilibrium (BCE)*—the solution concept most naturally associated with the problem of information design. We will also show that it is possible, in certain cases, to steer agents toward particular equilibria in an *online* manner, that is, *compute* the optimal equilibrium *while* steering players toward it.

### 12.2 Summary of Our Results

Here we summarize our model and results. There is a fixed, arbitrary extensive-form game  $\Gamma$ , being played repeatedly over rounds  $t = 1, \dots, T$ . Players’ rewards are assumed to be normalized to range  $[0, 1]$ . The players are assumed to play in such a way that their regret increases sublinearly as a function of  $T$ . This is a fairly natural and mild assumption (as discussed in the previous paragraph), and moreover there are many well-known algorithms that players can use to efficiently achieve sublinear regret in extensive-form games, perhaps the best-known of which is *counterfactual regret minimization* (Zinkevich et al., 2007), which has regret  $T^{1/2}$  ignoring game-dependent constants.<sup>62</sup>

<sup>62</sup>Throughout the introduction, game-dependent constants are omitted for clarity and to emphasize the dependence on  $T$ . In all cases, the omitted game-dependent constant is polynomial in the number of nodes in the game tree.





**Figure 29:** *Left:* An extensive-form version of a stag hunt. Chance plays uniformly at random at the root node, and the dotted line connecting the two nodes of Player 2 indicates an infoset: Player 2 cannot distinguish the two nodes. The game has two equilibria: one at the bottom-left corner, and one at the top-right corner (star). The latter is Pareto-dominant. Introducing vanishing realized payments alters the gradient landscape, steering players to the optimal equilibrium (star) instead of the suboptimal one (opposite corner). The capital letters show the players’ initial strategies. Lighter color indicates higher welfare and the star shows the highest-welfare equilibrium. Further details are in the appendix of the full paper (Zhang et al., 2024b).

Broadly speaking, the goal of our paper is to design methods of *steering* the learning behavior of the players so that they reach desirable equilibria instead of undesirable ones. We do this by introducing a *mediator* to the game. After each round, the mediator observes how the players played the game, and has the power to give nonnegative *payments*  $p_i^{(t)}$  to each player  $i$  at each round  $t$ . We will first consider the case where a target *pure Nash equilibrium* is given as part of the problem instance.

A few observations follow easily. If the mediator’s payments are not bounded, the mediator can trivially steer the players toward any outcome at all—not just equilibrium outcomes—by simply paying the players to play that outcome. We must therefore somehow bound the budget of the mediator. We will study two different budgets: a *per-round budget*, which constrains the individual payments  $p_i^{(t)}$ , and a *total budget*, which constrains their sum over time. We start by showing that the total budget must be allowed to grow with time.

**Proposition 12.1** (Informal version of Proposition 12.10). *For any fixed total budget  $B$ , there is a time horizon  $T$  large enough that the steering problem is impossible.*

As a result, the total budget must be allowed to grow with the time horizon, but yet, for the problem to be interesting, the budget cannot be allowed to grow too fast. We thus focus on the regime where the budget is allowed to grow with  $T$ , but only *sublinearly*—that is, the *average* per-round payment must vanish in the limit  $T \rightarrow \infty$ . We are interested in algorithms for which both the average budget and rate of convergence to the desired equilibrium can both be bounded by  $T^{-c}$  for some absolute constant  $c > 0$ . We show the following.

**Theorem 12.2** (Informal version of Theorem 12.12). *Steering to pure-strategy equilibria is possible in normal-form games, with absolute constant per-round budget. The average budget and rate of convergence to equilibrium are both  $T^{-1/4}$ .*

Intuitively, the mediator sends payments in such a way as to 1) reward the player a small amount for playing the equilibrium, and 2) *compensate* the player for deviations of other players. The goal of the mediator is to set the payments in such a way that the target equilibrium actions become *strictly dominant* for the players, and therefore the players must play them.

Next we turn to the extensive-form setting. Settings such as information design, in which first a signal is designed, and then players take actions, are naturally extensive-form games. We distinguish between two settings: the *full feedback* setting, in which the mediator observes every player’s entire strategy at every round,

and the *trajectory-feedback setting*, in which the mediator only observes the trajectories that are actually played by the players.<sup>63</sup>

The *full feedback* setting yields results similar to the normal-form setting.

**Theorem 12.3** (Informal version of Theorem 12.14). *Steering to pure-strategy equilibria is possible in extensive-form games with full feedback, with absolute constant per-round budget. The average budget and rate of convergence to equilibrium are both  $T^{-1/4}$ .*

The *trajectory feedback* case, however, is quite different.

**Theorem 12.4** (Informal version of Theorem 12.16). *With only trajectory feedback and absolute constant per-round budget, steering in general extensive-form games is impossible, even to the welfare-maximizing pure Nash equilibrium.*

Intuitively, the discrepancy is because, with only trajectory feedback, it is not possible to make the target equilibrium dominant using only nonnegative, vanishing-on-average payments, so the techniques used for the previous results cannot apply. This phenomenon can already be observed in the “stag hunt” game in Figure 29: for Player 2, Stag (S) cannot be a weakly-dominant strategy unless a payment is given at the boxed node, which would be problematic because such payments would also appear in the welfare-optimal equilibrium (S, S). Thus, one needs to be more clever. Fortunately, steering is still possible in this setting, but only if the per-round budget is also allowed to grow:

**Theorem 12.5** (Informal version of Theorem 12.14). *Steering to pure-strategy equilibria is possible in extensive-form games with full feedback. The average budget and rate of convergence to equilibrium are both  $T^{-1/8}$ , and the per-round budget grows at rate  $T^{1/8}$ .*

Next, we generalize our results beyond pure Nash equilibria. To do this, we will require the mediator to have the additional ability to give *advice* to the players, in the form of action recommendations. First, we show that using advice is a *necessary* condition for steering to even mixed Nash equilibria.

**Theorem 12.6** (Informal version of Theorem 12.19). *Without advice, there exists a normal-form game in which the unique optimal Nash equilibrium is mixed, and it is impossible to steer players toward it.*

If we allow advice, it turns out to be possible to steer players not just to mixed Nash equilibria but to a far broader set of equilibria known as the *Bayes-correlated equilibria*.

**Theorem 12.7** (Informal version of Theorem 12.21). *With advice, steering to Bayes-correlated equilibria is possible in extensive-form games. The conditions and rates are the same as those for pure Nash equilibria.*

Intuitively, the result follows because Bayes-correlated equilibria can be viewed as the pure Nash equilibria of an *augmented game* in which the advice is treated as part of the game’s observations. Bayes-correlated equilibria are a very general solution concept that include, for example, all the extensive-form correlated equilibria (von Stengel and Forges, 2008) and communication equilibria (Forges, 1986; Myerson, 1986), among other notions.

Finally, we give an *online* version of our algorithm, which does not need to know the target equilibrium beforehand. Instead, given an objective function, the online steering algorithm *steers players toward the optimal equilibrium while computing it*.

---

<sup>63</sup>This distinction becomes only meaningful for extensive-form games. For normal-form games, the two settings above coincide, because the “trajectory” in a normal-form game is just a list consisting of each player’s chosen action.

	Steering to Fixed Equilibrium	Online Steering
Normal Form or Full Feedback	$T^{-1/4}$ (Theorem 12.14)	$T^{-1/6}$ (Theorem 12.25)
Extensive Form and Trajectory Feedback	$T^{-1/8}$ (Theorem 12.18)	<i>Open problem</i>

**Table 30:** Summary of our positive algorithmic results. We hide game-dependent constants and logarithmic factors, and assume that regret minimizers incur regret  $T^{-1/2}$ .

**Theorem 12.8** (Informal version of Theorem 12.25). *In the full-feedback setting with advice and absolute constant per-round budget, it is possible to learn the optimal equilibrium while simultaneously steering the players toward it. The average budget and rate of convergence to equilibrium are both  $T^{-1/6}$ .*

As before, in normal-form games, full feedback and trajectory feedback essentially coincide, so online steering also turns out to be possible in normal-form games with trajectory feedback. In extensive-form games, however, the problem of trajectory-feedback online steering seems more difficult, and we leave it as an open problem. We summarize the rates we obtain in Section 12.2.

Finally, we complement our theoretical analysis by implementing and testing our steering algorithms in several benchmark games in Section 12.7.

## 12.3 The Steering Problem

In this section, we introduce what we call the *steering* problem. Informally, the steering problem asks whether a mediator can always steer players to any given equilibrium of an extensive-form game.

**Definition 12.9** (Steering Problem for Pure-Strategy Nash Equilibrium). Let  $\Gamma$  be an extensive-form game with payoffs bounded in  $[0, 1]$ . Let  $\mathbf{o}$  be an arbitrary pure-strategy Nash equilibrium of  $\Gamma$ , which we will call the *target equilibrium*. The mediator knows the game  $\Gamma$ , as well as a function  $R(T) = o(T)$ , which may be game-dependent, that bounds the regret of all players. At each round  $t \in [T]$ , the mediator picks *payment functions* for each player,  $p_i^{(t)} : \text{co } \mathcal{X}_1 \times \cdots \times \text{co } \mathcal{X}_n \rightarrow [0, P]$ , where  $p_i^{(t)}$  is linear in  $\mathbf{x}_i$  and continuous in  $\mathbf{x}_{-i}$ , and  $P$  defines the largest allowable per-iteration payment. Then, players pick strategies  $\mathbf{x}_i^{(t)} \in \mathcal{X}_i$ . Each player  $i$  then gets utility  $v_i^{(t)}(\mathbf{x}_i) := u_i(\mathbf{x}_i, \mathbf{x}_{-i}^{(t)}) + p_i^{(t)}(\mathbf{x}_i, \mathbf{x}_{-i}^{(t)})$ . The mediator has two desiderata.

(S1) (Payments) The time-averaged realized payments to the players, defined as

$$\max_{i \in [n]} \frac{1}{T} \sum_{t=1}^T p_i^{(t)}(\mathbf{x}^{(t)}),$$

converges to 0 as  $T \rightarrow \infty$ .

(S2) (Target Equilibrium) Players' actions are indistinguishable from the Nash equilibrium  $\mathbf{o}$ . That is, for every terminal node  $z$ , the *directness gap*, defined as

$$\sum_{z \in \mathcal{Z}} \left| \frac{1}{T} \sum_{t=1}^T \hat{\mathbf{x}}^{(t)}[z] - \hat{\mathbf{o}}[z] \right| = \left\| \frac{1}{T} \sum_{t=1}^T \hat{\mathbf{x}}^{(t)} - \hat{\mathbf{o}} \right\|_1,$$

converges to 0 as  $T \rightarrow \infty$ .

The assumption imposed on the payment functions in Definition 12.9 ensures the existence of Nash equilibria in the payment-augmented game (e.g., Fudenberg and Tirole, 1991, p. 34). Throughout this paper, we will refer to players as *direct* if they are playing actions prescribed by the target equilibrium strategy  $\mathbf{o}$ . Critically, (S2) does not require that the strategies themselves converge to the direct strategies, i.e.,  $\mathbf{x}_i^{(t)} \rightarrow \mathbf{o}_i$ , in iterates or in averages. They may differ on nodes off the equilibrium path. Instead, the requirement defined

by (S2) is that the *outcome distribution over terminal nodes* converges to that of the equilibrium. Similarly, (S1) refers to the *realized payments*  $p_i^{(t)}(\mathbf{x}^{(t)})$ , not the *maximum offered payment*  $\max_{\mathbf{x} \in \mathcal{X}} p_i^{(t)}(\mathbf{x})$ .

For now, we assume that the pure Nash equilibrium is part of the instance, and therefore our only task is to steer the agents toward it. In Section 12.6 we show how our steering algorithms can be extended to other equilibrium concepts such as *mixed* or (*Bayes-*)*correlated* equilibria, and to the case where the mediator needs to compute the equilibrium.

The mediator does not know anything about how the players pick their strategies, except that they will have regret bounded by a function that vanishes in the limit and is known to the mediator. This condition is a commonly adopted behavioral assumption (Nekipelov et al., 2015; Kolumbus and Nisan, 2022; Camara et al., 2020). The regret of Player  $i \in [n]$  in this context is defined as

$$\text{REG}_{\mathcal{X}_i}^T := \frac{1}{P+1} \left[ \max_{\mathbf{x}_i^* \in \mathcal{X}_i} \sum_{t=1}^T v_i^{(t)}(\mathbf{x}_i^*) - \sum_{t=1}^T v_i^{(t)}(\mathbf{x}_i^{(t)}) \right].$$

That is, regret takes into account the payment functions offered to that player. (The division by  $1/(P+1)$  is for normalization, since  $v_i^{(t)}$ 's has range  $[0, P+1]$ .)

How large payments are needed to achieve (S1) and (S2)? If the mediator could provide totally unconstrained payments, it could enforce any arbitrary outcome. On the other hand, if the total payments are restricted to be bounded, the steering problem is information-theoretically impossible:

**Proposition 12.10.** *There exists a game and some function  $R(T) = O(\sqrt{T})$  such that, for all  $B \geq 0$ , the steering problem is impossible if we add the constraint  $\sum_{t=1}^{\infty} \sum_{i=1}^n p_i^{(t)}(\mathbf{x}^{(t)}) \leq B$ .*

Hence, a weaker requirement on the size of the payments is needed. Between these extremes, one may allow the *total* payment to be unbounded, but insist that the *average* payment per round must vanish in the limit.

## 12.4 Steering in Normal-Form Games

We start with the simpler setting of *normal-form games*, that is, extensive-form games in which every player has one information set, and the set of histories correspond precisely to the set of pure profiles. This setting is much simpler than the general extensive-form setting (we consider in the next section), and we can appeal to a special case of a result in the literature Monderer and Tennenholtz (2004).

**Proposition 12.11** (Costless implementation of pure Nash equilibria, special case of  $k$ -implementation, Monderer and Tennenholtz, 2004). *Let  $\mathbf{o}$  be a pure Nash equilibrium in a normal-form game. Then there exist functions  $p_i^* : \text{co } \mathcal{X}_1 \times \cdots \times \text{co } \mathcal{X}_n \rightarrow [0, 1]$ , with  $p_i^*(\mathbf{o}) = 0$ , such that in the game with utilities  $v_i := u_i + p_i^*$ , the profile  $\mathbf{o}$  is weakly dominant:  $v_i(\mathbf{o}_i, \mathbf{x}_{-i}) \geq v_i(\mathbf{x}_i, \mathbf{x}_{-i})$  for every profile  $\mathbf{x}$ .*

The proof is constructive. The payment function

$$p_i^*(\mathbf{x}) := (\mathbf{o}_i^\top \mathbf{x}_i) \left( 1 - \prod_{j \neq i} \mathbf{o}_j^\top \mathbf{x}_j \right),$$

which on pure profiles  $\mathbf{x}$  returns 1 if and only if  $\mathbf{x}_i = \mathbf{o}_i$  and  $\mathbf{x}_j \neq \mathbf{o}_j$  for some  $j \neq i$  makes equilibrium play weakly dominant. It is *almost* enough for steering: the only problem is that  $\mathbf{o}$  is only *weakly* dominant, so no-regret players *may* play other strategies than  $\mathbf{o}$ . This can be fixed by adding a small reward  $\alpha \ll 1$  for playing  $\mathbf{o}_i$ . That is, we set

$$p_i(\mathbf{x}) := \alpha \mathbf{o}_i^\top \mathbf{x}_i + p_i^*(\mathbf{x}) = (\mathbf{o}_i^\top \mathbf{x}_i) \left( \alpha + 1 - \prod_{j \neq i} \mathbf{o}_j^\top \mathbf{x}_j \right). \quad (14)$$

On a high level, the structure of the payment function guarantees that the average strategy of any no-regret learner  $i \in [n]$  should be approaching the direct strategy  $\mathbf{o}_i$  by making  $\mathbf{o}_i$  the strictly dominant strategy of player  $i$ . At the same time, it is possible to ensure that the average payment will also be vanishing by appropriately selecting parameter  $\alpha$ . With an appropriate choice of  $\alpha$ , this is enough to solve the steering problem for normal-form games:

**Theorem 12.12** (Normal-form steering). *Let  $p_i(\mathbf{x})$  be defined as in (14), set  $\alpha = \sqrt{\epsilon}$ , where  $\epsilon := 4nR(T)/T$ , and let  $T$  be large enough that  $\alpha \leq 1$ . Then players will be steered toward equilibrium, with both payments and directness gap bounded by  $2\sqrt{\epsilon}$ .*

We note that no effort was made throughout this paper to optimize the game-dependent or constant factors, so long as they remained polynomial in  $|\mathcal{Z}|$ —they can very likely be improved.

## 12.5 Steering in Extensive-Form Games

This section considers steering in extensive-form games. We will first consider a model in which steering payments can condition on full player strategies (Section 12.5.1). Next, we consider a model in which only realized trajectories are considered (Section 12.5.2).

There are two main reasons why the extensive-form version of the steering problem is significantly more challenging than the normal-form version.

First, in extensive form, the strategy spaces of the players are no longer simplices. Therefore, if we wanted to write a payment function  $p_i$  with the property that  $p_i(\mathbf{x}) = \alpha \mathbb{1}\{\mathbf{x} = \mathbf{o}\} + \mathbb{1}\{\mathbf{x}_i = \mathbf{o}_i; \exists j \mathbf{x}_j \neq \mathbf{o}_j\}$  for pure  $\mathbf{x}$  (which is what was needed by Theorem 12.12), such a function would not be linear (or even convex) in player  $i$ 's strategy  $\mathbf{x}_i \in \text{co } \mathcal{X}_i$  (which is a sequence-form strategy, not a distribution over pure strategies). As such, even the meaning of extensive-form regret minimization becomes suspect in this setting.

Second, in extensive form, a desirable property would be that the mediator give payments conditioned only on what actually happens in gameplay, *not* on the players' full strategies—in particular, if a particular information set is not reached during play, the mediator should not know what action the player *would have* selected at that information set. We will call this the *trajectory* setting, and distinguish it from the *full-feedback* setting, where the mediator observes the players' full strategies.<sup>64</sup> This distinction is meaningless in the normal-form setting: since terminal nodes in normal form correspond to (pure) profiles, observing gameplay is equivalent to observing strategies. (We will discuss this point in more detail when we introduce the trajectory-feedback setting in Section 12.5.2.)

### 12.5.1 Steering with Full Feedback

In this section, we introduce a steering algorithm for extensive-form games under full feedback, summarized below.

**Definition 12.13** (FULLFEEDBACKSTEER). At every round, set the payment function  $p_i(\mathbf{x}_i, \mathbf{x}_{-i})$  as

$$\underbrace{\alpha \mathbf{o}_i^\top \mathbf{x}_i}_{\text{directness bonus}} + \underbrace{[u_i(\mathbf{x}_i, \mathbf{o}_{-i}) - u_i(\mathbf{x}_i, \mathbf{x}_{-i})]}_{\text{sandboxing payments}} - \underbrace{\min_{\mathbf{x}'_i \in \mathcal{X}_i} [u_i(\mathbf{x}'_i, \mathbf{o}_{-i}) - u_i(\mathbf{x}'_i, \mathbf{x}_{-i})]}_{\text{payment to ensure nonnegativity}}, \quad (15)$$

where  $\alpha \leq 1/|\mathcal{Z}|$  is a hyperparameter that we will select appropriately.

By construction,  $p_i$  satisfies the conditions of the steering problem (Definition 12.9): it is linear in  $\mathbf{x}_i$ , continuous in  $\mathbf{x}_{-i}$ , nonnegative, and bounded by an absolute constant (namely, 3). The payment function defined above has three terms:

<sup>64</sup>To be clear, the settings are differentiated by what the *mediator* observes, not what the *players* observe. That is, it is valid to consider the full-feedback steering setting with players running bandit-feedback regret minimizers, or the trajectory-feedback steering setting with players running full-feedback regret minimizing algorithms.

1. The first term is a *reward for directness*: a player gets a reward proportional to  $\alpha$  if it plays  $\mathbf{o}_i$ .
2. The second term *compensates the player* for the indirectness of other players. That is, the second term ensures that players' rewards are *as if* the other players had acted directly.
3. The final term simply ensures that the overall expression is nonnegative.

We claim that this protocol solves the basic version of the steering problem, as formalized below.

**Theorem 12.14.** *Set  $\alpha = \sqrt{\epsilon}$ , where  $\epsilon := 4nR(T)/T$ , and let  $T$  be large enough that  $\alpha \leq 1/|\mathcal{Z}|$ . Then, FULLFEEDBACKSTEER results in average realized payments and directness gap at most  $3|\mathcal{Z}|\sqrt{\epsilon}$ .*

## 12.5.2 Steering with Trajectory Feedback

In FULLFEEDBACKSTEER, payments depend on full strategies  $\mathbf{x}$ , not the realized game trajectories. In particular, the mediator in Theorem 12.14 observes what the players *would have played* even at infosets that other players avoid. To allow for an algorithm that works without knowledge of full strategies,  $p_i^{(t)}$  must be structured so that it could be induced by a payment function that only gives payments for terminal nodes reached during play. To this end, we now formalize *trajectory-feedback steering*.

**Definition 12.15** (Trajectory-feedback steering problem). Let  $\Gamma$  be an extensive-form game in which rewards are bounded in  $[0, 1]$  for all players. Let  $\mathbf{o}$  be an arbitrary pure-strategy Nash equilibrium of  $\Gamma$ . The mediator knows  $\Gamma$  and a regret bound  $R(T) = o(T)$ . At each  $t \in [T]$ , the mediator selects a payment function  $q_i^{(t)} : \mathcal{Z} \rightarrow [0, P]$ . The players select strategies  $\mathbf{x}_i^{(t)}$ . A terminal node  $z^{(t)} \sim \mathbf{x}^{(t)}$  is sampled, and all agents observe the terminal node that was reached,  $z^{(t)}$ . The players get payments  $q_i^{(t)}(z^{(t)})$ , so that their expected payment is  $p_i^{(t)}(\mathbf{x}) := \mathbb{E}_{z \sim \mathbf{x}} q_i^{(t)}(z)$ . The desiderata are as in Definition 12.9.

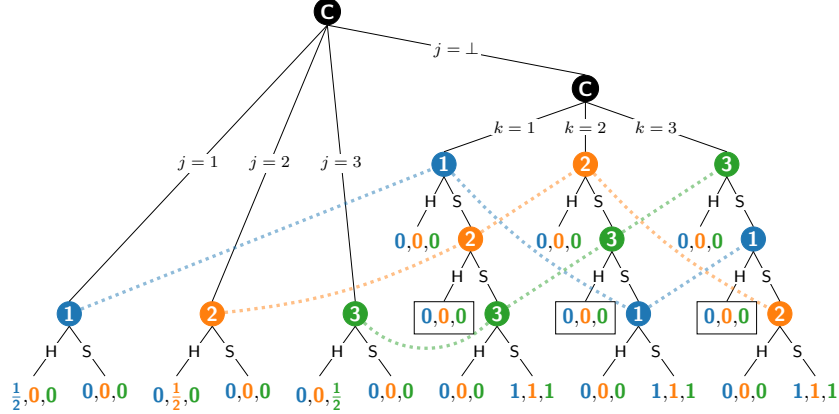
The trajectory-feedback steering problem is more difficult than the full-feedback steering problem in two ways. First, as discussed above, the mediator does not observe the strategies  $\mathbf{x}$ , only a terminal node  $z^{(t)} \sim \mathbf{x}$ . Second, the form of the payment function  $q_i^{(t)} : \mathcal{Z} \rightarrow [0, P]$  is restricted: this is already sufficient to rule out FULLFEEDBACKSTEER. Indeed,  $p_i$  as defined in (15) cannot be written in the form  $\mathbb{E}_{z \sim \mathbf{x}} q_i(z)$ :  $p_i(\mathbf{x}_i, \mathbf{x}_{-i})$  is nonlinear in  $\mathbf{x}_{-i}$  due to the nonnegativity-ensuring payments, whereas every function of the form  $\mathbb{E}_{z \sim \mathbf{x}} q_i(z)$  will be linear in each player's strategy.

We remark that, despite the above algorithm containing a sampling step, the payment function is defined *deterministically*: the payment is defined as the *expected value*  $p_i^{(t)}(\mathbf{x}) := \mathbb{E}_{z \sim \mathbf{x}} q_i^{(t)}(z)$ . Thus, the theorem statements in this section will also be deterministic.

In the normal-form setting, the payments  $p_i$  defined by (14) already satisfy the condition of trajectory-feedback steering. In particular, if  $z$  is the terminal node, we have

$$p_i(\mathbf{x}) = \mathbb{E}_{z \sim \mathbf{x}} [\alpha \mathbb{1}\{z = z^*\} + \mathbb{1}\{\mathbf{x}_i = \mathbf{o}_i; \exists j \mathbf{x}_j \neq \mathbf{o}_j\}].$$

Therefore, in the normal-form setting, Theorem 12.12 applies to both full-feedback steering and trajectory-feedback steering, and we have no need to distinguish between the two. However, in extensive form, as discussed above, the two settings are quite different.



**Figure 31:** The counterexample for Theorem 12.16, for  $n = 3$ . Chance always plays uniformly at random. Infosets are linked by dotted lines (all nodes belonging to the same player are in the same info set).

### 12.5.3 Lower Bound

Unlike in the full-feedback or normal-form settings, in the trajectory-feedback setting, steering is impossible in the general case in the sense that per-iteration payments bounded by any constant do not suffice.

**Theorem 12.16.** For every  $P > 0$ , there exists an extensive-form game  $\Gamma$  with  $O(P)$  players,  $O(P^2)$  nodes, and rewards bounded in  $[0, 1]$  such that, with payments  $q_i^{(t)} : \mathcal{Z} \rightarrow [0, P]$ , it is impossible to steer players to the welfare-maximizing Nash equilibrium, even when  $R(T) = 0$ .

For intuition, consider the extensive-form game in Figure 31, which can be seen as a three-player version of Stag Hunt. Players who play Hare (H) get a value of  $1/2$  (up to constants); in addition, if all three players play Stag (S), they all get expected value 1. The welfare-maximizing equilibrium is “everyone plays Stag”, but “everyone plays Hare” is also an equilibrium. In addition, if all players are playing Hare, the only way for the mediator to convince a player to play Stag without accidentally also paying players in the Stag equilibrium is to pay players at one of the three boxed nodes. But those three nodes are only reached with probability  $1/n$  as often as the three nodes on the left, so the mediator would have to give a bonus of more than  $n/2$ . The full proof essentially works by deriving an algorithm that the players could use to exploit this dilemma to achieve either large payments or bad convergence rate, generalizing the example to  $n > 3$ , and taking  $n = \Theta(P)$ . The formal proof is deferred to the full paper (Zhang et al., 2024b).

### 12.5.4 Upper Bound

To circumvent the lower bound in Theorem 12.16, in this subsection, we allow the payment bound  $P \geq 1$  to depend on both the time limit  $T$  and the game.

**Definition 12.17** (TRAJECTORYSTEER). Let  $\alpha, P$  be hyperparameters. Then, for all rounds  $t = 1, \dots, T$ , sample  $z \sim \mathbf{x}^{(t)}$  and pay players as follows. If all players have been direct (i.e., if  $\hat{o}[z] = 1$ ), pay all players  $\alpha$ . If at least one player has not been direct, pay  $P$  to all players who have been direct. That is, set  $q_i^{(t)}(z^{(t)}) = \alpha \hat{o}[z] + P \mathbf{o}_i[z](1 - \hat{o}[z])$ .

**Theorem 12.18.** Set the hyperparameters  $\alpha = 4|\mathcal{Z}|^{1/2}\epsilon^{1/4}$  and  $P = 2|\mathcal{Z}|^{1/2}\epsilon^{-1/4}$ , where  $\epsilon := R(T)/T$ , and let  $T$  be large enough that  $\alpha \leq 1$ . Then, running TRAJECTORYSTEER for  $T$  rounds results in average realized payments bounded by  $8|\mathcal{Z}|^{1/2}\epsilon^{1/4}$ , and directness gap by  $2\epsilon^{1/2}$ .

As alluded to in the introduction, the proof of this result is more involved than those for previous results,

because one cannot simply make the target equilibrium dominant as in the full-feedback case. One may hope that—as in FULLFEEDBACKSTEER—the desired equilibrium can be made dominant by adding payments. In fact, a sort of “chicken-and-egg” problem arises: (S2) requires that all players converge to equilibrium. But for this to happen, other players’ strategies must first converge to equilibrium so that  $i$ ’s incentives are as they would be in equilibrium. The main challenge in the proof of Theorem 12.18 is therefore to carefully set the hyperparameters to achieve convergence despite these apparent problems.

## 12.6 Other Equilibrium Notions and Online Steering

So far, Theorems 12.14 and 12.18 handle only the case where the equilibrium is a *pure-strategy* Nash equilibrium of the game, given as part of the input. This section extends our analysis to other equilibrium notions and considers settings in which an *objective for the mediator* is given instead of a target equilibrium. For the former, we will show that many types of equilibrium can be viewed as pure-strategy equilibria in an *augmented game* in which the mediator has the ability to give *advice* to the players in the form of action recommendations. Then, in the original game, the goal is to guide the players to the pure strategy profile of following recommendations.

### 12.6.1 Necessity of Advice

We first show that without the possibility to give advice, steering is impossible with sublinear payments.

**Theorem 12.19.** *There exists a normal-form game, and objective function  $u_M$  of the mediator, such that the unique optimal equilibrium is mixed, and it is impossible to steer players toward that equilibrium using only sublinear payments (and no advice).*

Given this result, we will analyze a setting in which the mediator is allowed to provide “advice,” and show a broad possibility result for steering.

### 12.6.2 More General Equilibrium Notions: Bayes-Correlated Equilibrium

Throughout this subsection, there will be two games: the original game  $\hat{\Gamma}$ , and the augmented game  $\Gamma$ . We will use hats to distinguish the various components of them. For example, a history of  $\hat{\Gamma}$  is  $\hat{h} \in \hat{\mathcal{H}}$ , a strategy of Player  $i$  is  $\hat{x}_i \in \hat{\mathcal{X}}_i$ , and so on. Given an  $n$ -player game  $\hat{\Gamma}$ , the *mediator-augmented game*  $\Gamma$  is the  $n + 1$ -player game constructed as follows.  $\Gamma$  is identical to  $\hat{\Gamma}$ , except that there is an extra player, namely, the mediator itself. We will denote the mediator as Player 0. For each (non-chance) player  $i$ , every decision point  $\hat{h} \in \hat{\mathcal{H}}_i$  is replaced with the following gadget. First, the mediator selects an action  $\hat{a} \in \hat{A}(\hat{h})$  to *recommend* to Player  $i$ . Player  $i$  privately observes the recommendation, and only then is allowed to choose an action. The mediator is assumed to have perfect information in the game. To ensure that the size of  $\Gamma$  is not too large, we make the following restriction: once two players have disobeyed action recommendations (“deviated”), the mediator ceases to give further action recommendations. Finally, upon reaching a terminal node  $\hat{z} \in \hat{\mathcal{Z}}$ , each player gets utility  $\hat{u}_i(\hat{z})$ .

We first analyze the size of  $\Gamma$ . A terminal node in  $\Gamma$  can be uniquely identified by a tuple  $(\hat{z}, \hat{h}_1, \hat{h}_2, \hat{a}_1, \hat{a}_2)$  where  $\hat{z}$  is the terminal node in the original game that was reached,  $\hat{h}_1, \hat{h}_2$  are predecessors of  $\hat{z}$  at which players deviated (or  $\emptyset$  if the deviations did not happen), and  $\hat{a}_1$  and  $\hat{a}_2$  are the recommendations that the mediator gave at  $\hat{h}_1, \hat{h}_2$  respectively (again,  $\emptyset$  if the deviations did not happen). Thus, a (very loose) bound on the number of terminal nodes in  $\Gamma$  is  $|\mathcal{Z}| \leq |\hat{\mathcal{Z}}|^3$ , *i.e.*, it is polynomial. (This is where we use the fact that only two deviations were allowed.)

As in the previous section, the mediator is able to *commit* to a strategy  $\mu \in \text{co } \mathcal{X}_M$  upfront on each iteration. For a fixed mediator strategy  $\mu$ , we will use  $\Gamma^\mu$  to refer to the  $n$ -player game resulting from treating the mediator as a nature player that plays according to  $\mu$ .

The *direct strategy*  $\mathbf{o}_i \in \mathcal{X}_i$  of each player  $i$  is the strategy that follows all mediator recommendations. The goal of the mediator is to find a *Bayes-correlated equilibrium*, which is defined as follows.



**Definition 12.20.** A *Bayes-correlated equilibrium*  $\Gamma$  is a strategy  $\boldsymbol{\mu} \in \text{co}\mathcal{X}_M$  for the mediator such that  $\boldsymbol{o}$  is a Nash equilibrium of  $\Gamma^\mu$ . An equilibrium  $\boldsymbol{\mu}$  is *optimal* if, among all equilibria, it maximizes the mediator’s objective  $u_M(\boldsymbol{\mu}, \boldsymbol{o})$ .

Bayes-correlated equilibria (BCEs) were introduced first by [Bergemann and Morris \(2016\)](#) in single-step games. In sequential (extensive-form) games, BCEs were explored first, to our knowledge, by [Makris and Renou \(2023\)](#) in the economics literature, and in independent work in the computer science literature as a special case of the general framework introduced by [Zhang and Sandholm \(2022a\)](#). Bayes-correlated equilibria are easily seen to be a superset of most other equilibrium notions, including (mixed) Nash equilibria, *extensive-form correlated equilibria* (EFCE) ([von Stengel and Forges, 2008](#)), *communication equilibria* ([Forges, 1986](#); [Myerson, 1986](#)), and many more. The *revelation principle* assures us that the assumption that players will be direct in equilibrium is without loss of generality: for every possible Nash equilibrium  $\boldsymbol{x}$  of  $\Gamma^\mu$ , then there is some  $\boldsymbol{\mu}'$  such that  $u_i(\boldsymbol{\mu}', \boldsymbol{o}) = u_i(\boldsymbol{\mu}, \boldsymbol{x})$ .

BCEs naturally capture the problems of *information design* and *Bayesian persuasion* (e.g., [Kamenica and Gentzkow \(2011\)](#)). In particular, the results in this section can therefore be thought of as a version of information design/Bayesian persuasion that does not need to assume that players will play a certain profile  $(\boldsymbol{o})$ , but instead *steers* the players to play that profile.

Since  $\Gamma^\mu$  is just an  $n$ -player game with pure Nash equilibrium  $\boldsymbol{o}$ , all of the results in the previous sections apply. Therefore, it follows immediately that is possible to steer players toward *any* BCE (and thus any mixed Nash equilibrium, any EFCE, or any communication equilibrium) so long as the mediator is allowed to give advice to the players. We therefore have the following result.

**Theorem 12.21.** *Algorithms FULLFEEDBACKSTEER and TRAJECTORYSTEER can be used to steer players to an arbitrary Bayes-correlated equilibrium, with (up to a polynomial loss in the dependence on  $|\hat{\mathcal{Z}}|$ , because  $|\mathcal{Z}| = \text{poly}(|\hat{\mathcal{Z}}|)$ ) the same bounds.*

### 12.6.3 Online Steering

We now consider the setting where the target equilibrium is *not* given to us beforehand. We assume that the mediator wishes to steer players toward an *optimal* equilibrium, but does not *a priori* know what that optimal equilibrium is. Instead of a target Nash equilibrium, we assume that the mediator has a utility function  $\hat{u}_M : \hat{\mathcal{Z}} \rightarrow [0, 1]$ , and we will call  $\hat{u}_M$  the *objective*. As with players’ utility functions,  $\hat{u}_M$  in  $\hat{\Gamma}$  induces a mediator utility function  $u_M$  in  $\Gamma$ . In particular, we would like to steer players toward an *optimal* equilibrium  $\boldsymbol{\mu}$ , without knowing that equilibrium beforehand. To that end, we add a new criterion.

- (S3) (Optimality) The mediator’s reward should converge to the reward of the optimal equilibrium. That is, the *optimality gap*  $u_M^* - \frac{1}{T} \sum_{t=1}^T u_M(\boldsymbol{\mu}^{(t)}, \boldsymbol{x}^{(t)})$ , where  $u_M^*$  is the mediator utility in an optimal equilibrium, converges to 0 as  $T \rightarrow \infty$ .

Since equilibria in mediator-augmented games are just strategies  $\tilde{\boldsymbol{\mu}}$  under which  $\tilde{\boldsymbol{o}}$  is a Nash equilibrium, we may use the following algorithm to steer players toward an optimal Bayes-correlated equilibrium:

**Definition 12.22** (COMPUTETHENSTEER). Compute an optimal equilibrium  $\boldsymbol{\mu}$ . With  $\boldsymbol{\mu}$  held fixed, run any steering algorithm in  $\Gamma^\mu$ .

As observed earlier, the main weakness of COMPUTETHENSTEER is that it must compute an equilibrium offline. Although this can be done in polynomial time ([Zhang and Sandholm, 2022a](#)), it is still far less efficient than, for example, a single step of a regret minimizer. To sidestep this, in this section we will introduce algorithms that compute the equilibrium in an *online* manner, while steering players toward it. Our algorithms will make use of a Lagrangian dual formulation analyzed by [Zhang et al. \(2023a\)](#).

**Proposition 12.23** (Zhang et al. (2023a)). *There exists a (game-dependent) constant  $\lambda^* \geq 0$  such that, for every  $\lambda \geq \lambda^*$ , the solutions  $\boldsymbol{\mu}$  to*

$$\max_{\boldsymbol{\mu} \in \text{co } \mathcal{X}_M} \min_{\boldsymbol{x}_i \in \text{co } \mathcal{X}_i; i \in [n]} u_M(\boldsymbol{\mu}, \boldsymbol{o}) - \lambda \sum_{i=1}^n [u_i(\boldsymbol{\mu}, \boldsymbol{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{\mu}, \boldsymbol{o}_i, \boldsymbol{o}_{-i})], \quad (16)$$

*are exactly the optimal equilibria of the augmented game.*

**Definition 12.24** (ONLINESTEER). The mediator runs a regret minimization algorithm  $\mathcal{R}_M$  over its own strategy space  $X_M$ , which we assume has regret at most  $R_M(T)$  after  $T$  rounds. On each round, the mediator does the following:

- Get a strategy  $\boldsymbol{\mu}^{(t)}$  from  $\mathcal{R}_M$ . Play  $\boldsymbol{\mu}^{(t)}$ , and set  $p_i^{(t)}$  as defined in (15) in  $\Gamma^{\boldsymbol{\mu}^{(t)}}$ .
- Pass utility  $\boldsymbol{\mu} \mapsto \frac{1}{\lambda} u_M(\boldsymbol{\mu}, \boldsymbol{o}) - \sum_{i=1}^n [u_i(\boldsymbol{\mu}, \boldsymbol{x}_i^{(t)}, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{\mu}, \boldsymbol{o}_i, \boldsymbol{o}_{-i})]$  to  $\mathcal{R}_M$ , where  $\lambda \geq 1$  is a hyperparameter.

**Theorem 12.25.** *Set the hyperparameters  $\alpha = \epsilon^{2/3} |\mathcal{Z}|^{-1/3}$  and  $\lambda = |\mathcal{Z}|^{2/3} \epsilon^{-1/3}$ , where  $\epsilon := (R_M(T) + 4nR(T))/T$  is the average regret bound summed across players, and let  $T$  be large enough that  $\alpha \leq 1/|\mathcal{Z}|$ . Then running ONLINESTEER results in average realized payments, directness gap, and optimality gap all bounded by  $7\lambda^* |\mathcal{Z}|^{4/3} \epsilon^{1/3}$ .*

The argument now works with the zero-sum formulation (16), and leverages the fact that the agents' average strategies are approaching the set of Nash equilibria since they have vanishing regrets. Thus, each player's average strategy should be approaching the direct strategy, which in turn implies that the average utility of the mediator is converging to the optimal value, analogously to Theorem 12.14. We provide the formal argument in the full paper (Zhang et al., 2024b)

It is worth noting that, despite the fact that it would speed up the convergence, we cannot set  $\lambda$  and  $\alpha$  dependent on  $\lambda^*$ , because we do not know  $\lambda^*$  a priori.

Algorithm ONLINESTEER can also be used to steer to optimal equilibria in other notions of equilibrium, such as *communication equilibrium* (Forges, 1986; Myerson, 1986), by using appropriate constructions of mediator-augmented games. The Bayes-correlated equilibrium is the most natural and general of these notions, so it is the one we use in our paper. For a more general discussion of mediator-augmented games, see Zhang and Sandholm (2022a).

ONLINESTEER has a further guarantee that FULLFEEDBACKSTEER does not, owing to the fact that it learns an equilibrium online: it works even when the players' sets of deviations,  $X_i$ , is not known upfront. In particular, the following generalization of Theorem 12.25 follows from an identical proof.

**Corollary 12.26.** *Suppose that each player  $i$ , unbeknownst to the mediator, is choosing from a subset  $\mathcal{Y}_i \subseteq \mathcal{X}_i$  of strategies that includes the direct strategy  $\boldsymbol{o}_i$ . Then, running Theorem 12.25 with the same hyperparameters yields the same convergence guarantees, except that the mediator's utility converges to its optimal utility against the true deviators, that is, a solution to (16) with each  $\mathcal{X}_i$  replaced by  $\mathcal{Y}_i$ .*

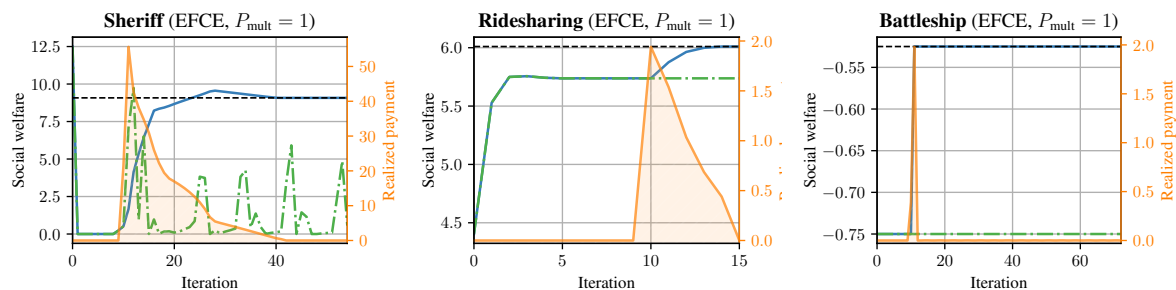
At this point, it is very reasonable to ask whether it is possible to perform *online* steering with *trajectory* feedback. In *normal-form* games, as with offline setting, there is minimal difference between the trajectory- and full-feedback settings. This intuition carries over to the trajectory-feedback setting: ONLINESTEER can be adapted into an online trajectory-feedback steering algorithm for normal-form games, with essentially the same convergence guarantee. We defer the formal statement of the algorithm and proof to the appendix of the full paper (Zhang et al., 2024b).

The algorithm, however, fails to extend to the *extensive-form* online trajectory-feedback setting, for the same reasons that the *offline* full-feedback algorithm fails to extend to the online setting. We leave extensive-form

online trajectory-feedback steering as an interesting open problem.

## 12.7 Experimental Results

We ran experiments with our TRAJECTORYSTEER algorithm (Definition 12.17) on various notions of equilibrium in extensive-form games, using the COMPUTETHENSTEER framework suggested by Definition 12.22. Since the hyperparameter settings suggested by Definition 12.17 are very extreme, in practice we fix a constant  $P$  and set  $\alpha$  dynamically based on the currently-observed gap to directness. We used CFR+ (Tammelin, 2014) as the regret minimizer for each player, and precomputed a welfare-optimal equilibrium with the LP algorithm of Zhang and Sandholm (2022a). In most instances tested, a small constant  $P$  (say,  $P \leq 8$ ) is enough to steer CFR+ regret minimizers to the exact equilibrium in a finite number of iterations. Two plots exhibiting this behavior are shown in Figure 32. More experiments, as well as descriptions of the game instances tested, can be found in the appendix of the full paper (Zhang et al., 2024b).



**Figure 32:** Sample experimental results. The blue line in each figure is the social welfare (left y-axis) of the players with steering enabled. The green dashed line is the social welfare without steering. The yellow line gives the payment (right y-axis) paid to each player. The flat black line denotes the welfare of the optimal equilibrium. The panels show the game, the equilibrium concept (in this figure, always EFCE). In all cases, the first ten iterations are a “burn-in” period during which no payments are issued; steering only begins after that.

## 12.8 Conclusions and Future Research

We established that it is possible to steer no-regret learners to optimal equilibria using vanishing rewards, even under trajectory feedback. There are many interesting avenues for future research. First, is there a natural *trajectory-feedback, online* algorithm that combines the desirable properties of both ONLINESTEER and TRAJECTORYSTEER? Second, this paper did not attempt to provide optimal rates, and their improvement is a fruitful direction for future work. Third, are there algorithms with less demanding knowledge assumptions for the principal, *e.g.*, steering without knowledge of utility functions? Finally, our main behavioral assumption throughout this paper is that the regret players incur vanishes in the limit. Yet, stronger guarantees could be possible when specific no-regret learning dynamics are in place, such as mean-based learning (Braverman et al., 2018); see (Vlatakis-Gkaragkounis et al., 2020; Giannou et al., 2021a,b) for recent results in the presence of *strict* equilibria. Concretely, it would be interesting to understand the class of learning dynamics under which the steering problem can be solved with a finite cumulative budget.

## Part IV

# Subgame Solving in Large Games

## 13 Subgame Solving without Common Knowledge

### 13.1 Introduction

*Subgame solving* is the standard technique for playing perfect-information games that has been used by strong agents in a wide variety of games, including chess (Campbell et al., 2002; Stockfish) and go (Silver et al., 2016). Methods for subgame solving in perfect-information games exploit the fact that a solution to a subgame can be computed independently of the rest of the game. However, this condition fails in the imperfect-information setting, where the optimal strategy in a subgame can depend on strategies outside that subgame.

Recently, subgame solving techniques have been extended to imperfect-information games (Ganzfried and Sandholm, 2015a; Jackson, 2014). Some of those techniques are provably *safe* in the sense that, under reasonable conditions, incorporating them into an agent cannot make the agent more exploitable (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017; Moravčík et al., 2017; Brown et al., 2018; Šustr et al., 2019; Brown et al., 2020; Kovařík et al., 2021). These techniques formed the core ingredient toward recent superhuman breakthroughs in AIs for no-limit Texas hold'em poker (Brown and Sandholm, 2018, 2019b). However, all of the prior techniques have a shared weakness that limits their applicability: as a first step, they enumerate the entire *common-knowledge closure* of the player's current info set, which is the smallest set of states within which it is common knowledge that the current node lies. In two-player community-card poker (in which each player is dealt private hole cards, and all actions are public, e.g., Texas hold'em), for example, the common-knowledge closure contains one node for each assignment of hole cards to both players. This set has a manageable size in such poker games, but in other games, it is unmanageably large.

We introduce a different technique to avoid having to enumerate the entire common-knowledge closure. We enumerate only the set of nodes corresponding to  $k$ th-order knowledge for finite  $k$ —in the present work, we focus mostly on the case  $k = 1$ , for it already gives us interesting results. This allows an agent to only conduct subgame solving on still-reachable states, which in general is a much smaller set than the whole common-knowledge subgame.

We prove that, as is, the resulting algorithm, 1-KLSS, does not guarantee safety, but we develop three avenues by which safety can be guaranteed. First, safety is guaranteed if the results of subgame solves are incorporated back into the blueprint strategy. Second, we provide a method by which safety is achieved by limiting the info sets at which subgame solving is performed. Third, we prove that our approach, when applied at every info set reached during play, achieves a weaker notion of equilibrium, which we coin *affine equilibrium* and which may be of independent interest. We show that affine equilibria cannot be exploited by any Nash strategy of the opponent: an opponent who wishes to exploit an affine equilibrium must open herself to counter-exploitation. Even without these three safety-guaranteeing additions, experiments on medium-sized games show that 1-KLSS always reduced exploitability in practical games even when applied at every info set.

We use depth-limited 1-KLSS to create, to our knowledge, the first agent capable of playing *dark chess*, a large imperfect-information variant of chess with similar game tree size, at a high level. We test it against opponents of various levels, including a baseline agent, an amateur-level human, and the world's highest-rated player. Our agent defeated the former two handily, and, despite losing to the top human, exhibited strong performance in the opening and midgame, often gaining a significant advantage before losing it in the endgame.

## 13.2 Preliminaries

In this section, we consider *timeable* two-player zero-sum games of imperfect recall with explicitly-defined observations. That is, each player has a function  $o_i : \mathcal{H} \rightarrow \mathbb{R}$  defining the observation that player  $i$  makes at history  $h$ . The sequence  $s_i(h)$  consists of all observations made by the player at nodes up to and including  $h$ . The set of sequences of player  $i$  is  $\Sigma_i$ .

We say that two states  $h = \emptyset a_1 \dots a_t$  and  $h' = \emptyset b_1 \dots b_t$  are *indistinguishable to player  $i$* , denoted  $h \sim_i h'$ , if  $s_i(h) = s_i(h')$ . An equivalence class of nodes  $h \in \mathcal{H}$  under  $\sim_i$  is an infoset. Notice that infosets are well-defined here even for the player not moving—this will be critical later on.

If  $u, u'$  are nodes or sequences,  $u \preceq u'$  means  $u$  is an ancestor of  $u'$  (or  $u' = u$ ). If  $S$  is a set of nodes,  $h \succeq S$  means  $h \succeq h'$  for some  $h' \in S$ , and  $\bar{S} = \{z : z \succeq S\}$ .

A *sequence-form mixed strategy* (hereafter *strategy*) of player  $i$  is a vector  $\mathbf{x} \in \mathbb{R}^{\Sigma_i}$ , in which  $\mathbf{x}[s]$  denotes the probability that player  $i$  plays all the actions in the sequence  $s$ . If  $h$  is a node or infoset, then we will use the overloaded notation  $\mathbf{x}[h] := \mathbf{x}[s_i(h)]$ .

The *counterfactual best-response value* (hereafter *best-response value*)  $u^*(\mathbf{x}|Ia)$  to a  $\blacktriangle$ -strategy  $\mathbf{x} \in \text{co}\mathcal{X}$  upon playing action  $a$  at  $I$  is the normalized best value for  $\blacktriangledown$  against  $\mathbf{x}$  after playing  $a$  at  $I$ :  $u^*(\mathbf{x}|Ia) = \frac{1}{\sum_{h \in I} p(h)\mathbf{x}[h]} \min_{\mathbf{y} \in \mathcal{Y}: \mathbf{y}[Ia]=1} \sum_{z: s_{\blacktriangledown}(z) \succeq Ia} u(z)p(z)\mathbf{x}[z]\mathbf{y}[z]$ . The best-response value at an infoset  $I$  is defined as  $u^*(\mathbf{x}|I) = \max_a u^*(\mathbf{x}|Ia)$ . The *best-response value*  $u^*(\mathbf{x})$  (without specifying an infoset) is the best-response value at the root, i.e.,  $\min_{\mathbf{y} \in \text{co}\mathcal{Y}} u(\mathbf{x}, \mathbf{y})$ . Analogous definitions hold for  $\blacktriangledown$ -strategy  $\mathbf{y}$  and  $\blacktriangle$ -infoset  $I$ .

We say that two nodes  $h$  and  $h'$  are *transpositions* if an observer who begins observing the game at  $h$  or  $h'$  and sees both players' actions and observations at every timestep cannot distinguish between the two nodes. Formally,  $h, h'$  are transpositions if, for all action sequences  $a_1 \dots a_t$ :

1.  $ha_1 \dots a_t$  is valid (i.e., for all  $j$ ,  $a_j$  is a legal move in  $ha_1 \dots a_{j-1}$ ) if and only if  $h'a_1 \dots a_t$  is valid, and in this case, we have  $o_i(ha_1 \dots a_j) = o_i(h'a_1 \dots a_j)$  for all players  $i$  and times  $0 \leq j \leq t$ , and
2.  $ha_1 \dots a_t$  is terminal if and only if  $h'a_1 \dots a_t$  is terminal, and in this case, we have  $u(ha_1 \dots a_t) = u(h'a_1 \dots a_t)$ .

For example, ignoring draw rules, two chess positions are transpositions if they have equal piece locations, castling rights, and *en passant* rights.

## 13.3 Common-Knowledge Subgame Solving

In this section we discuss prior work on subgame solving. First,  $\blacktriangle$  computes a blueprint strategy  $\mathbf{x}$  for the full game. During a playthrough,  $\blacktriangle$  reaches an infoset  $I$ , and would like to perform subgame solving to refine her strategy for the remainder of the game. All prior subgame solving methods that we are aware of require, as a first step, constructing (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017; Moravčík et al., 2017; Brown et al., 2018; Šustr et al., 2019; Brown et al., 2020; Kovařík et al., 2021), or at least approximating via samples (Šustr et al., 2021), the *common-knowledge closure* of  $I$ .

**Definition 13.1.** The *infoset hypergraph*  $\mathcal{G}$  of a game  $\Gamma$  is the hypergraph whose vertices are the nodes of  $\Gamma$ , and whose hyperedges are information sets.

**Definition 13.2.** Let  $S$  be a set of nodes in  $\Gamma$ . The *order- $k$  knowledge set*  $S^k$  is the set of nodes that are at most distance  $k - 1$  away from  $S$  in  $\mathcal{G}$ . The *common-knowledge closure*  $S^\infty$  is the connected component of  $\mathcal{G}$  containing  $S$ .

Intuitively, if we know that the true node is in  $S$ , then we know that the opponent knows that the true node is in  $S^2$ , we know that the opponent knows that we know that the true node is in  $S^3$ , etc., and it is common knowledge that the true node is in  $S^\infty$ . After constructing  $I^\infty$  (where  $I$ , as above, is the infoset  $\blacktriangle$  has reached), standard techniques then construct the subgame  $\bar{I}^\infty$  (or an abstraction of it), and solve it to obtain the refined strategy. In this section we describe three variants: *resolving* (Burch et al., 2014), *maxmargin* (Moravcik et al., 2016), and *reach subgame solving* (Brown and Sandholm, 2017).

Let  $H_{\text{top}}$  be the set of root nodes of  $I^\infty$ , that is, the set of nodes  $h \in I^\infty$  for which the parent of  $h$  is not in  $I^\infty$ . In *subgame resolving*, the following gadget game is constructed. First, nature chooses a node  $h \in H_{\text{top}}$  with probability proportional to  $p(h)\mathbf{x}[h]$ . Then,  $\blacktriangledown$  observes her info set  $I_{\blacktriangledown}(h)$ , and is given the choice to either *exit* or *play*. If she exits, the game ends at a terminal node  $z$  with  $u(z) = u^*(\mathbf{x}|I_{\blacktriangledown}(h))$ . This payoff is called the *alternate payoff* at  $I_{\blacktriangledown}(h)$ . Otherwise, the game continues from node  $h$ . In *maxmargin* solving, the objective is changed to instead find a strategy  $\mathbf{x}'$  that maximizes the minimum *margin*  $M(I) := u^*(\mathbf{x}'|I) - u^*(\mathbf{x}|I)$  associated with any  $\blacktriangledown$ -info set  $I$  intersecting  $H_{\text{top}}$ . (Resolving only ensures that all margins are positive). This can be accomplished by modifying the gadget game. In *reach subgame solving*, the alternative payoffs  $u^*(\mathbf{x}|I)$  are decreased by the *gift* at  $I$ , which is a lower bound on the magnitude of error that  $\blacktriangledown$  has made by playing to reach  $I$  in the first place. Reach subgame solving can be applied on top of either resolving or maxmargin.

The full game  $\Gamma$  is then replaced by the gadget game, and the gadget game is resolved to produce a strategy  $\mathbf{x}'$  that  $\blacktriangle$  will use to play to play after  $I$ . To use nested subgame solving, the process repeats when another new info set is reached.

## 13.4 Knowledge-Limited Subgame Solving

In this section we introduce the main contribution of our paper, *knowledge-limited subgame solving*. The core idea is to reduce the computational requirements of safe subgame solving methods by discarding nodes that are “far away” (in the info set hypergraph  $\mathcal{G}$ ) from the current info set.

Fix an odd positive integer  $k$ . In *order- $k$  knowledge-limited subgame solving* ( *$k$ -KLSS*), we fix  $\blacktriangle$ 's strategy outside  $\overline{I^k}$ , and then perform subgame solving as usual. Pseudocode for all algorithms can be found in the appendix. This carries many advantages:

1. Since  $\blacktriangle$ 's strategy is fixed outside  $\overline{I^k}$ ,  $\blacktriangledown$ 's best response outside  $\overline{I^{k+1}}$  is also fixed. Thus, all nodes outside  $\overline{I^{k+1}}$  can be pruned and discarded.
2. At nodes  $h \in \overline{I^{k+1}} \setminus \overline{I^k}$ ,  $\blacktriangle$ 's strategy is again fixed. Thus, the payoff at these nodes is only a function of  $\blacktriangledown$ 's strategy in the subgame and the blueprint strategy. These payoffs can be computed from the blueprint and added to the row of the payoff matrix corresponding to  $\blacktriangle$ 's empty sequence. These nodes can then also be discarded, leaving only  $\overline{I^k}$ .
3. Transpositions can be accounted for if  $k = 1$  and we allow a slight amount of incorrectness. Suppose that  $h, h' \in I$  are transpositions. Then  $\blacktriangle$  cannot distinguish  $h$  from  $h'$  ever again. Further,  $\blacktriangledown$ 's information structure after  $h$  in  $\overline{I^k}$  is identical to her information structure in  $h'$  in  $\overline{I^k}$ . Thus, in the payoff matrix of the subgame,  $h$  and  $h'$  induce two disjoint sections of the payoff matrix  $A_h$  and  $A_{h'}$  that are identical except for the top row (thanks to Item 2 above). We can thus remove one (say, at random) without losing too much. If one section of the matrix contains entries that are all not larger than the corresponding entries of the other part, then we can remove the latter part without any loss since it is weakly dominated.

The transposition merging may cause incorrect behavior (over-optimism) in games such as poker, but we believe that its effect in a game like dark chess, where information is transient at best and the evaluation of a position depends more on the actual position than on the players' information, is minor. Other abstraction techniques can also be used to reduce the size of the subgame, if necessary. We will denote the resulting gadget game  $\Gamma[I^k]$ .

In games like dark chess, even individual info sets can have size  $10^7$ , which means even  $I^2$  can have size  $10^{14}$  or larger. This is wholly unmanageable in real time. Further, very long shortest paths can exist in the info set hypergraph  $\mathcal{G}$ . As such, it may be difficult to even determine whether a given node is in  $I^\infty$ , much less expand all its nodes, even approximately. Thus, being able to reduce to  $I^k$  for finite  $k$  is a large step in making subgame solving techniques practical.

The benefit of KLSS can be seen concretely in the following parameterized family of games which we coin  *$N$ -matching pennies*. We will use it as a running example in the rest of the paper. Nature first chooses an integer  $n \in \{1, \dots, N\}$  uniformly at random.  $\blacktriangle$  observes  $\lfloor n/2 \rfloor$  and  $\blacktriangledown$  observes  $\lfloor (n+1)/2 \rfloor$ . Then,  $\blacktriangle$  and

▼ simultaneously choose heads or tails. If they both choose heads, ▲ scores  $n$ . If they both choose tails, ▲ scores  $N - n$ . If they choose opposite sides, ▲ scores 0. For any infoset  $I$  just after nature makes her move, there is no common knowledge whatsoever, so  $\bar{I}^\infty$  is the whole game except for the root nature node. However,  $I^k$  consists of only  $\Theta(k)$  nodes.

On the other hand, in community-card poker,  $I^\infty$  itself is quite small: indeed, in heads-up Texas Hold’Em,  $I^\infty$  always has size at most  $\binom{52}{2} \cdot \binom{50}{2} \approx 1.6 \times 10^6$  and even fewer after public cards have been dealt. Furthermore, game-specific tricks or matrix sparsification (Johanson et al., 2011; Zhang and Sandholm, 2020b) can make game solvers behave as if  $I^\infty \approx 10^3$ . This is manageable in real time, and is the key that has enabled recent breakthroughs in AIs for no-limit Texas hold’em (Moravčík et al., 2017; Brown and Sandholm, 2018, 2019b). In such settings, we do not expect our techniques to give improvement over the current state of the art.

The rest of this section addresses the *safety* of KLSS. The techniques in Section 13.3 are *safe* in the sense that applying them at every infoset reached during play in a nested fashion cannot increase exploitability compared to the blueprint strategy (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017). KLSS is not safe in that sense:

**Proposition 13.3.** *There exists a game and blueprint for which applying 1-KLSS at every infoset reached during play increases exploitability by a factor linear in the size of the game.*

Despite the above negative example, we now give multiple methods by which we can obtain safety guarantees when using KLSS.

### 13.4.1 Safety by Updating the Blueprint

Our first method of obtaining safety is to immediately and permanently update the blueprint strategy after every subgame solution is computed. Proofs of the results in this section can be found in the appendix.

**Theorem 13.4.** *Suppose that whenever  $k$ -KLSS is performed at infoset  $I$  (e.g., it can be performed at every infoset reached during play in a nested manner), and that subgame strategy is immediately and permanently incorporated into the blueprint, thereby overriding the blueprint strategy in  $\bar{I}^k$ . Then the resulting sequence of blueprints has non-increasing exploitability.*

To recover a full safety guarantee from Theorem 13.4, the blueprint—not the subgame solution—should be used during play, and the only function of the subgame solve is to update the blueprint for later use. One way to track the blueprint updates is to store the computed solutions to all subgames that the agent has ever solved. In games where only a reasonably small number of paths get played in practice (this can depend on the strength and style of the players), this is feasible. In other games this might be prohibitively storage intensive.

It may seem unintuitive that we cannot use the subgame solution on the playthrough on which it is computed, but we can use it forever after that (by incorporating it into the blueprint), while maintaining safety. This is because, if we allow the *choice of information set*  $I$  in Theorem 13.4 to depend on the opponent’s strategy, the resulting strategy is exploitable due to Proposition 13.3. By only using the subgame solve result at later playthroughs, the choice of  $I$  no longer depends on the opponent strategy at the later playthrough, so we recover a safety guarantee.

One might further be concerned that what the opponent or nature does in some playthrough of the game affects our strategy in later playthroughs and thus the opponent can learn more about, or affect, the strategy she will face in later playthroughs. However, this is not a problem. If the blueprint is an  $\epsilon$ -NE, the opponent (or nature) can affect *which*  $\epsilon$ -NE we will play at later playthroughs, but because we will always play from *some*  $\epsilon$ -NE, we remain unexploitable.

In the rest of this section we prove forms of safety guarantees for 1-KLSS that do not require the blueprint to be updated at all.

### 13.4.2 Safety by Allocating Deviations from the Blueprint

We now show that another way to achieve safety of 1-KLSS is to carefully allocate how much it is allowed to deviate from the blueprint. Let  $\mathcal{G}'$  be the graph whose nodes are infosets for  $\blacktriangle$ , and in which two infosets  $I$  and  $I'$  share an edge if they contain nodes that are in the same  $\blacktriangledown$ -infoset. In other words,  $\mathcal{G}'$  is the infoset hypergraph  $\mathcal{G}$ , but with every  $\blacktriangle$ -infoset collapsed into a single node.

**Theorem 13.5.** *Let  $\mathbf{x}$  be an  $\epsilon$ -NE blueprint strategy for  $\blacktriangle$ . Let  $\mathcal{I}$  be an independent set in  $\mathcal{G}'$  that is closed under ancestor (that is, if  $I \succeq I'$  and  $I \in \mathcal{I}$ , then  $I' \in \mathcal{I}$ ). Suppose that 1-KLSS is performed at every infoset in  $\mathcal{I}$ , to create a strategy  $\mathbf{x}'$ . Then  $\mathbf{x}'$  is also an  $\epsilon$ -NE strategy.*

To apply this method safely, we may select beforehand a distribution  $\pi$  over independent sets of  $\mathcal{G}'$ , which induces a map  $p : V(\mathcal{G}') \rightarrow \mathbb{R}$  where  $p(I) = \Pr_{\mathcal{I} \sim \pi}[I \in \mathcal{I}]$ . Then, upon reaching infoset  $I$ , with probability  $1 - p(I)$ , play the blueprint until the end of the game; otherwise, run 1-KLSS at  $I$  (possibly resulting in more nested subgame solves) and play that strategy instead. It is always safe to set  $p(I) \leq 1/\chi(I^\infty)$  where  $\chi(I^\infty)$  denotes the chromatic number of the subgraph of  $\mathcal{G}'$  induced by the infosets in the common-knowledge closure  $I^\infty$ . For example, if the game is perfect information, then  $\mathcal{G}'[I^\infty]$  is the trivial graph with only one node  $I$ , so, as expected, it is safe to set  $p(I) = 1$ , that is, perform subgame solving everywhere.

### 13.4.3 Affine Equilibrium, which Guarantees Safety against All Equilibrium Strategies

We now introduce the notion of *affine equilibrium*. We will show that such equilibrium strategies are safe against all NE strategies, which implies that they are only exploitable by playing non-NE strategies, that is, by opening oneself up to counter-exploitation. We then show that 1-KLSS finds such equilibria.

**Definition 13.6.** A vector  $\mathbf{x}$  is an *affine combination* of vectors  $x_1, \dots, x_k$  if  $\mathbf{x} = \sum_{i=1}^k \alpha_i x_i$  with  $\sum_i \alpha_i = 1$ , where the coefficients  $\alpha_i$  can have arbitrary magnitude and sign.

**Definition 13.7.** An *affine equilibrium strategy* is an affine combination of NE strategies.

In particular, if the NE is unique, then so is the affine equilibrium. Before stating our safety guarantees, we first state another fact about affine equilibria that illuminates their utility.

**Proposition 13.8.** *Every affine equilibrium is a best response to every NE strategy of the opponent.*

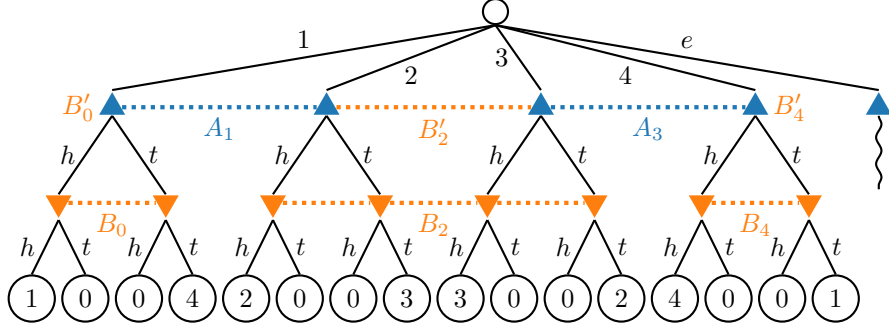
In other words, every affine equilibrium is an NE of the restricted game  $\Gamma'$  in which  $\blacktriangledown$  can only play her NE strategies in  $\Gamma$ . That is, affine equilibria are not exploitable by NE strategies of the opponent, not even by safe exploitation techniques (Ganzfried and Sandholm, 2015b). So, the only way for the opponent to exploit an affine equilibrium is to open herself up to counter-exploitation. Affine equilibria may be of independent interest as a reasonable relaxation of NE in settings where finding an exact or approximate NE strategy may be too much to ask for.

**Theorem 13.9.** *Let  $\mathbf{x}$  be a blueprint strategy for  $\blacktriangle$ , and suppose that  $\mathbf{x}$  happens to be an NE strategy. Suppose that we run 1-KLSS using the blueprint  $\mathbf{x}$ , at every infoset in the game, to create a strategy  $\mathbf{x}'$ . Then  $\mathbf{x}'$  is an affine equilibrium strategy.*

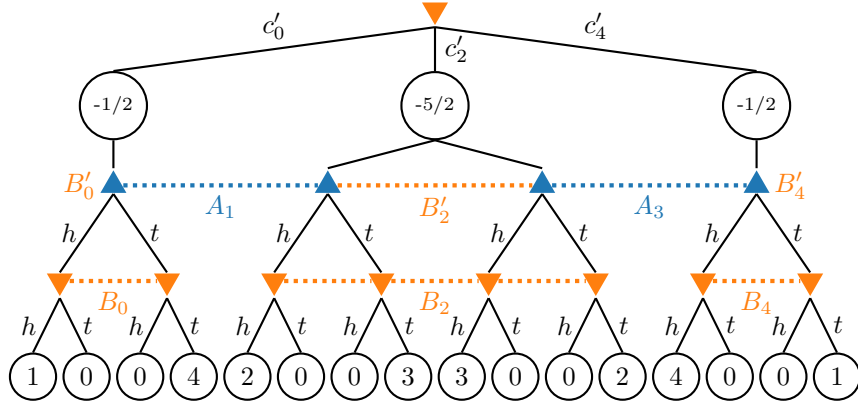
The theorem could perhaps be generalized to approximate equilibria, but the loss of a large factor (linear in the size of the game, in the worst case) in the approximation would be unavoidable: the counterexample in the proof of Proposition 13.3 has a  $\Theta(1/N)$ -NE becoming a  $\Theta(1)$ -NE, in a game where the Nash equilibria are already affine-closed (that is, all affine combinations of Nash equilibria are Nash equilibria). Furthermore, it is nontrivial to even define  $\epsilon$ -affine equilibrium.

Theorem 13.9 and Proposition 13.3 together suggest that 1-KLSS may make mistakes when  $\mathbf{x}$  suffers from *systematic* errors (e.g., playing a certain action  $a$  too frequently *overall* rather than in a particular infoset).





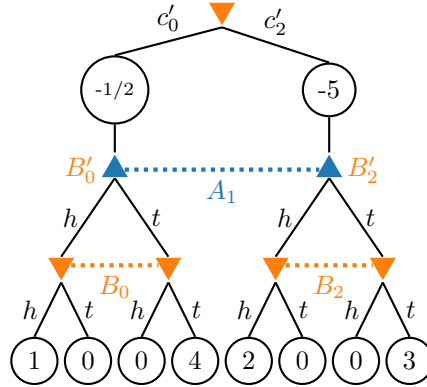
**Figure 33:** A simple game that we use in our example. The game is a modified version of 4-matching pennies. Blank nodes are nature or terminal; terminal nodes are labeled with their utilities. Nodes will be referred to by the sequence of edges leading to that node; for example, the leftmost terminal node is  $1hh$ . The details of the subgame at  $e$  are irrelevant. Nature’s strategy at the root node is uniform random.



**Figure 34:** The common-knowledge subgame at  $A_1$ ,  $\Gamma[A_1^\infty]$ . Nature’s strategy at all its nodes, once again, is uniform random. The nodes  $c'_0$  and  $c'_4$  are redundant because nature only has one action, but we include these for consistency with the pseudocode.

1-KLSS may overcorrect for such errors, as the counterexample clearly shows. Intuitively, if the blueprint plays action  $a$  too often (e.g., folds in poker), 1-KLSS may try to correct for that game-wide error fully in each infoset, thereby causing the strategy to overall be very far from equilibrium (e.g., folding way too infrequently in poker). However, we will demonstrate that this overcorrection never happens in our experiments in practical games, even if the blueprint contains very systematic errors.

Strangely, the proofs of both Theorem 13.9 and Theorem 13.5 do not work for  $k$ -KLSS when  $k > 1$ , because it is no longer the case that the strategies computed by subgame solving are necessarily played—in particular, for  $k > 1$ ,  $k$ -KLSS on an infoset  $I$  computes strategies for infosets  $I'$  that are no longer reachable, and such strategies may never be played. For  $k = \infty$ —that is, for the case of common knowledge—it is well known that the theorems hold via different proofs (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017). We leave the investigation of the case  $1 < k < \infty$  for future research.



**Figure 35:** The subgame for 1-KLSS at  $A_1$ . Once again, both nature nodes are redundant, but included for consistency with the pseudocode. The counterfactual value at  $c'_2$  is scaled up because the other half of the subtree is missing. In addition to this,  $\blacktriangledown$  gains value  $3/2$  for playing  $h$  and  $1$  for playing  $t$  at  $B_2$ , accounting for that missing subtree.

### 13.5 Example of How 1-KLSS Works

Figure 33 shows a small example game. Suppose that the  $\blacktriangle$ -blueprint is uniform random, and consider an agent who has reached infoset  $A_1$  and wishes to perform subgame solving. Under the given blueprint strategy,  $\blacktriangledown$  has the following counterfactual values:  $1/2$  at  $B'_0$  and  $C'_4$ , and  $5/2$  at  $B'_2$ .

The common-knowledge maxmargin gadget subgame  $\Gamma[A_1^\infty]$  can be seen in Figure 34. The 1-KLSS maxmargin gadget subgame  $\Gamma[A_1]$  can be seen in Figure 35.

The advantage of 1-KLSS is clearly demonstrated in this example: while both KLSS and common-knowledge subgame solving prune out the subgame at node 5, 1-KLSS further prunes the subgames at node 4 (because it is outside the order-2 set  $A_1^2$  and thus does not directly affect  $A_1$ ) and node 3 (because it only depends on  $\blacktriangledown$ 's strategy in the subgame—and not on  $\blacktriangle$ 's strategy—and thus can be added to a single row of  $B$ ).

The payoff matrices corresponding to these gadget subgames can be found in the appendix of the full paper (Zhang and Sandholm, 2021b).

### 13.6 Dark Chess: An Agent from Only a Value Function Rather Than a Blueprint

In this section, we give an overview of our dark chess agent, which uses 1-KLSS as a core ingredient. More details can be found in the appendix of the full paper (Zhang and Sandholm, 2021b). Although we wrote our agent in a game-specific fashion, many techniques in this section also apply to other games.

**Definition 13.10.** A *trunk* of a game  $\Gamma$  is a modified version of  $\Gamma$  in which some internal nodes  $h$  of  $\Gamma$  have been replaced by terminal nodes and given utilities. We will call such nodes *internal leaves*. When working with a trunk, internal leaves  $h$  can be *expanded* by adding all of their children into the tree, giving these children utilities, and removing the utility assigned to  $h$ .

In dark chess, constructing a blueprint is already a difficult problem due to the sheer size of the game, and expanding the whole game tree is clearly impractical. Instead, we resort to a *depth-limited* version of 1-KLSS. In depth-limited subgame solving, only a trunk of the game tree is expanded explicitly, and approximations are made to the leaves of the trunk.

Conventionally in depth-limited subgame solving of imperfect-information games, at each trunk leaf, both players are allowed to choose among *continuation strategies* for the remainder of the game (Brown et al., 2018, 2020; Kovařík et al., 2021; Šustr et al., 2021). In the absence of a mechanism for creating a reasonable blueprint, much less multiple blueprints to be used as continuation strategies, we resort to only using an

approximate value function  $\tilde{u} : \mathcal{H} \rightarrow \mathbb{R}$ . We will not formally define what a good value function is, except that it should roughly approximate “the value” of a node  $h \in \mathcal{H}$ , to the extent that such a quantity exists (for a more rigorous treatment of value functions in subgame solving, see Kovařík et al., 2021 (Kovařík et al., 2021)). In this setting, this is not too bothersome: the dominant term in any reasonable node-value function in dark chess will be material count, which is common knowledge anyway. We use a value function based on *Stockfish 13*, currently the strongest available chess engine.

Subgame solving in imperfect-information games with only approximate leaf values (and no continuation strategies) has not been explored to our knowledge (since it is not theoretically sound), but it seems reasonable to assume that it would work well with sufficient depth, since increasing depth effectively amounts to adding more and more continuation strategies.

To perform nested subgame solving, every time it is our turn, we perform 1-KLSS at our current information set. The generated subgame then replaces the original game, and the process repeats. This approach has the notable problem of information loss over time: since all the solves are depth-limited, eventually, we will reach a point where we fall off the end of the initially-created game tree. At this point, those nodes will disappear from consideration. From a game-theoretic perspective, this equates to always assuming that the opponent knew the exact state of the game  $d$  timesteps ago, where  $d$  is the search depth. As a remedy, one may consider sampling some number of infosets  $I' \succeq I^2 \setminus I$  to continue expanding. We do not investigate this possibility here, as we believe that it would not yield a significant performance benefit in dark chess (and may even hurt in practice: since no blueprint is available at  $I'$ , a new blueprint would have to be computed. This effectively amounts to 3-KLSS, which may lack theoretical guarantees compared to 1-KLSS).

## 13.7 Experiments

**Experiments in medium-sized games.** We conducted experiments on various small and medium-sized games to test the practical performance of 1-KLSS. To do this, we created a blueprint strategy for  $\blacktriangle$  that is intentionally weak by forcing  $\blacktriangle$  to play an  $\epsilon$ -uniform strategy (i.e., at every infoset  $I$ , every action  $a$  must be played with probability at least  $\epsilon/m$  where  $m$  is the number of actions at  $I$ ). The blueprint is computed as the least-exploitable strategy under this condition. During subgame solving, the same restriction is applied at every infoset except the root, which means theoretically that it is possible for any strategy to arise from nested solving applied to every infoset in the game. The mistakes made by playing with this restriction are highly systematic (namely, playing bad actions with positive probability  $\epsilon$ ); thus, the argument at the end of Section 13.4 suggests that we may expect order-1 subgame solving to perform poorly in this setting.

We tested on a wide variety of games, including some implemented in the open-source library *OpenSpiel* (Lancot et al., 2019). All games were solved with Gurobi 9.0 (Gurobi Optimization, LLC, 2020), and subgames were solved in a nested fashion at every information set using *maxmargin* solving. We found that, in all practical games (i.e., all games tested except the toy game 100-matching pennies) 1-KLSS in practice always decreases the exploitability of the blueprint, suggesting that 1-KLSS decreases exploitability in practice, despite the lack of matching theoretical guarantees. Experimental results can be found in Table 36. We also conducted experiments at  $\epsilon = 0$  (so that the blueprint is an exact NE strategy, and all the subgame solving needs to do is not inadvertently ruin the equilibrium), and found that, in all games tested, the equilibrium strategy was indeed not ruined (that is, exploitability remained 0). Gurobi was reset before each subgame solution was computed, to avoid warm-starting the subgame solution at equilibrium.

The experimental results suggest that despite the behavior of 1-KLSS in our counterexample to Proposition 13.3, in practice 1-KLSS can be applied at every infoset without increasing exploitability despite lacking theoretical guarantees.

**Experiments in dark chess.** We used our techniques to create an agent capable of playing dark chess. We tested on dark chess instead of other imperfect-information chess variants, such as *Kriegspiel* or *recon chess*, because dark chess has recently been implemented by a major chess website, chess.com (under the name *Fog of War Chess*), and has thus exploded in recent popularity, producing strong human expert players. Our agent runs on a single machine with 6 CPU cores.

We tested our agent by playing three different opponents:

game	exploitability		
	blueprint	after 1-KLSS	ratio
2x2 Abrupt Dark Hex	0.07	0.06	1.09
4-card Goofspiel, random order	0.17	0.08	2.2
4-card Goofspiel, increasing order	0.17	0.0	$\infty$
Kuhn poker	0.01	1.5	8.3
Kuhn poker ( $\epsilon$ -bet)	3.5	0.0	$\infty$
3-rank limit Leduc poker	0.02	0.02	1.09
3-rank limit Leduc poker ( $\epsilon$ -fold)	6.5	5.7	1.09
3-rank limit Leduc poker ( $\epsilon$ -bet)	9.7	9.6	1.01
Liar’s Dice, 5-sided die	0.18	0.13	1.45
100-Matching pennies	1.3	9.8	0.13

**Table 36:** *Experimental results in medium-sized games. Reward ranges in all games were normalized to lie in  $[-1, 1]$ . Ratio is the blueprint exploitability divided by the post-subgame-solving exploitability. The value  $\epsilon$  was set to 0.25 in all experiments, but the results are qualitatively similar with smaller values of  $\epsilon$  such as 0.1. In the  $\epsilon$ -bet/fold variants, the blueprint is the least-exploitable strategy that always plays that action with probability at least  $\epsilon$  (Kuhn poker with 0.25-fold has an exact Nash equilibrium for P1, so we do not include it). Descriptions and statistics about the games can be found in the appendix.*

1. A 100-game match against a baseline agent, which is, in short, the same algorithm as our agent, except that it only performs imperfect-information search to depth 1, and after that uses *Stockfish*’s perfect-information evaluation with iterative deepening. The baseline agent is described in more detail the appendix. Our agent defeated it by a score of 59.5–40.5, which is statistically significant at the 95% level.
2. One of the authors of this paper is rated approximately 1700 on chess.com in Fog of War, and has played upwards of 20 games against the agent, winning only two and losing the remainder.
3. Ten games against FIDE Master Luis Chan (“luizzy”), who is currently the world’s strongest player on the Fog of War blitz rating list<sup>65</sup> on chess.com, with a rating of 2416. Our agent lost the match 9–1. Despite the loss, our agent demonstrated strong play in the opening and midgame phases of the game, often gaining a large advantage before throwing it away in the endgame by playing too pessimistically.

The performances against the two humans put the rating of our agent at approximately 2000, which is a strong level of play. The agent also exhibited nontrivial plays such as bluffing by attacking with unprotected pieces, and making moves that exploit the opponent’s lack of knowledge—something that agents like the baseline agent could never do. We have compiled and uploaded some representative samples of gameplay of our dark chess agent, with comments, at [this link](#).

## 13.8 Conclusions and Future Research

We developed a novel approach to subgame solving,  $k$ -KLSS, in imperfect-information games that avoids dealing with the common-knowledge closure. Our methods vastly increase the applicability of subgame solving techniques; they can now be used in settings where the common-knowledge closure is too large to enumerate or approximate. We proved that as is, this does not guarantee safety of the strategy, but we developed three avenues by which safety guarantees can be achieved. First, safety is guaranteed if the results of subgame solves are incorporated back into the blueprint strategy. Second, the usual guarantee of safety against *any* strategy can be achieved by limiting the infosets at which subgame solving is performed. Third, we proved that 1-KLSS, when applied at every infoset reached during play, achieves a weaker notion of equilibrium, which we coin *affine equilibrium* and which may be of independent interest. We showed that affine equilibria cannot be exploited by any Nash strategy of the opponent, so an opponent who wishes to

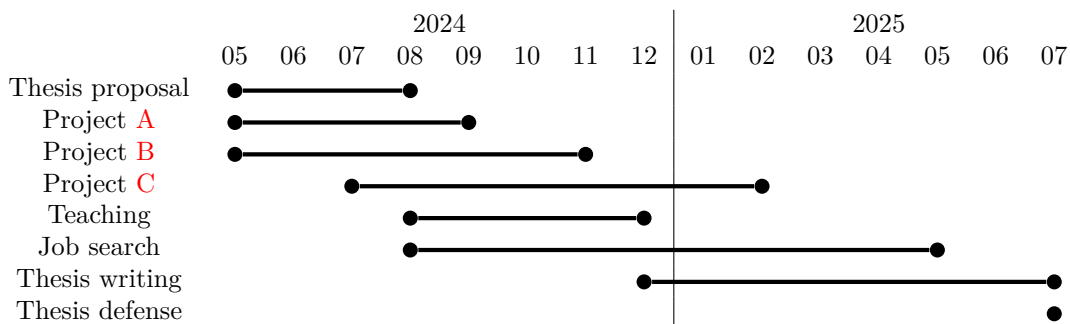
<sup>65</sup>That rating list is by far the most active, so it is reasonable to assume those ratings are most representative.

exploit an affine equilibrium must open herself to counter-exploitation. Even without the safety-guaranteeing additions, experiments on medium-sized games showed that 1-KLSS always reduced exploitability in practical games even when applied at every infoset, and depth-limited 1-KLSS led to, to our knowledge, the first strong AI for dark chess.

This opens many future research directions:

1. Analyze  $k$ -KLSS for  $1 < k < \infty$  in theory and practice.
2. Incorporate function approximation via neural networks to generate blueprints, particles, or both.
3. Improve techniques for large games such as dark chess, especially managing possibly-game-critical uncertainty about the opponent's position and achieving deeper, more accurate search.

# Future Research and Timeline



**Table 37:** Proposed timeline of work to be done between now and thesis defense. “Teaching” refers to the course 15-888 (Computational Game Solving), which I will co-teach with Tuomas Sandholm in the fall semester. “Job search” includes preparation, submission, and presentation of both applications (fall) and interviews (spring). Project D is a stretch goal, and I will spend time on it only if I have extra time to spare.

I hope to work on the following future directions and submit any attained results for publication by May 2025, and finish and defend my thesis in Summer 2025. The order of the items in this section is the rough order in which I plan to work on these topics, and Table 37 contains a rough proposed timeline.

## A Toward a Superhuman Dark Chess Agent

Joint work with Tuomas Sandholm.

We are currently investigating ways to further improve KLSS (described in Part IV) with the goal of achieving superhuman performance in dark chess. Since the publication of our paper (Zhang and Sandholm, 2021b), we have significantly improved the agent to the point that it currently achieves about an 86% score against the agent from our published paper. We plan to improve the performance as much as possible, and then test the agent via matches on the servers of chess.com, the same way that we tested the agent for our earlier paper. Here we detail some of the improvements that we have made.

### A.1 Learning-Based Game Solvers

Instead of using linear programming to solve the subgames as in Part IV, we switch to using CFR. Namely, we use a multithreaded implementation of *growing-tree* CFR (GT-CFR) (Schmid et al., 2023) with some, which we will now describe for completeness. GT-CFR has two components, running in parallel.

1. A *node expander*. The node expander repeatedly selects at random a leaf node using Algorithm `SelectNode`, and expands it.
2. A *game solver*, running `PCFR+`. It performs alternating iterates of `PCFR+`, on each iteration traversing only those nodes which have been expanded by the node expander.

We now elaborate in greater detail on each of these two components.

---

**Algorithm SelectNode**

---

```
1: EXPLORER  $\leftarrow$  randomly-chosen player
2:  $h \leftarrow$  root of current subgame
3: while  $h$  is not a leaf do
4:    $\triangleright$  avoid excess node expansions if current strategy is low-support
5:   if  $h$ 's infoset has not yet been touched by the game solver then return NULL
6:    $\mathbf{x}_h \in \Delta(A(h)) \leftarrow$  current game solver strategy at  $h$ 
7:    $\triangleright \mathbb{1}\{\mathbf{x}_h > 0\}$  is taken element-wise
8:   if EXPLORER plays at  $h$  then  $a \leftarrow$  sample action  $a$  w.p.  $(1 - \epsilon) \frac{\mathbb{1}\{\mathbf{x}_h(a) > 0\}}{|\text{supp } \mathbf{x}_h|} + \epsilon \frac{1}{|A(h)|}$ 
9:   else  $a \leftarrow$  sample from  $\mathbf{x}_h$ 
10:  if  $h$  is a chance node then
11:     $\triangleright$   $h$  is the root of a subgame, i.e., a node at which chance samples  $\blacktriangle$ 's information
12:     $\triangleright$  from the blueprint, given  $\blacktriangledown$ 's information. We should focus our attention on nodes in  $\blacktriangle$ 's
13:     $\triangleright$  current infoset, so spend at least half our node expansion budget expanding such nodes
14:    with probability 0.5 do  $a \leftarrow$  the action leading to  $\blacktriangle$ 's current infoset
15: return  $h$ 
```

---

### A.1.1 The Node Expander

The node expander selects and expands new nodes of the game tree. The node selection algorithm is given in Algorithm [SelectNode](#). Compared to [Schmid et al. \(2023\)](#), our algorithm uses a different method of traversing the tree. In particular, their method uses MCTS for node selection, whereas we use a *one-sided*,  $\epsilon$ -greedy approach, where:

- one player, the *explorer*, selects uniformly at random among the “best” actions, *i.e.*, the actions in the support of its current strategy, with some  $\epsilon$  noise for exploration, and
- the other player samples from its current strategy.

The “one-sided” nature of this exploration procedure ensures that nodes that are in neither players’ support are not needlessly explored. We also experimented with MCTS, which we found in practice to be inferior<sup>66</sup>.

The selected node  $h$  (if not NULL) is then *expanded*. That is,  $h$  is added to the infoset  $I(h)$  that contains it (creating this infoset in memory if needed), and all children of  $h$  are evaluated with Stockfish’s evaluation function and then added to the game tree with those values.

Several threads running the node expander algorithm are run in parallel. In particular, our current implementation uses 3 parallel node expander threads<sup>67</sup>.

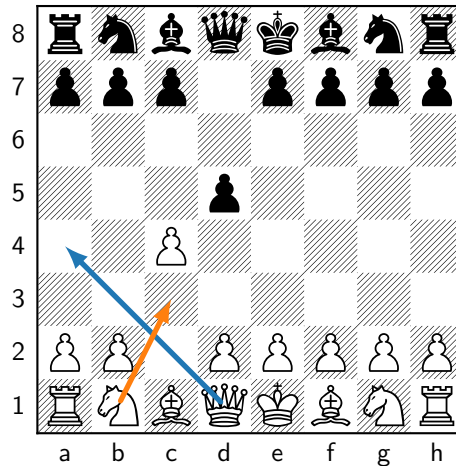
Our implementation is completely lock-free except that locks are required at each node to ensure that two different threads do not attempt to simultaneously expand the same node or create the same information set. Such locks are only required for the node expander, and therefore in particular the game solver is completely lock-free.

---

<sup>66</sup>This phenomenon may be of independent interest: if  $\epsilon$ -greedy outperforms MCTS in our setting, perhaps it may also be superior in more “traditional” settings, such as in two-player zero-sum games *a la AlphaZero* ([Silver et al., 2016](#)).

<sup>67</sup>Increasing the number of node expander threads does not appear to improve performance. We believe that this is because of some combination of lock contention and game solving being unable to keep up with the game tree growth.

**Figure 38:** Position after 1. c4 d5. White can win almost a full pawn (in expectation) by mixing between the moves 2. Qa4 with low probability and 2. Nc3 with high probability. No move for Black simultaneously defends the threats against both the king and the pawn. (2... c6 may look like it does, but after 3. cxd5, Black cannot recapture the pawn without risking hanging a king or queen).



### A.1.2 The Game Solver

The game solver performs alternating-iterate **PCFR+** on the current game tree and eventually outputs the last iterate. We avoid computing or storing the average iterate, and instead we simply compute and play according to the last iterate **PCFR+**. We do this for several reasons.

- Since the game tree is growing with time, it is not clear how to define the “average iterate” of **PCFR+**, since different infosets will have been in the tree for different amounts of time.
- **PCFR+** with alternating updates appears to exhibit last-iterate convergence in practice<sup>68</sup>. Thus, it is unnecessary to use the average iterate to achieve convergence.
- The entropy inherent to the algorithm (for example, in the choice of what node to expand), in addition to the stochasticity of the termination time (since the termination condition is a time limit, rather than a fixed number of iterates) means that the last iterate is essentially already random, thus removing the need for additional randomness.

Alternate values for use in subgame solving are computed as expected values of the final joint strategy, as in Schmid et al. (2023).

## A.2 Other Improvements

Here we discuss simple but significant details in our implementation.

**Purification.** After solving the subgame, we *purify* the strategy (Ganzfried et al., 2012). In particular, when we do not know the current state (*i.e.*, there is more than one current subgame root), we deterministically play the highest-probability action. Otherwise, we mix among the top three actions. This had the effect of reducing low-probability blunders that the agent plays too often due to the various approximations made throughout the agent, improving overall performance.

<sup>68</sup>Although this is not known in theory, in practice the last-iterate convergence is in fact fairly fast!



Purification will, of course, increase exploitability in general. In Figure 38, we give a very common opening example. This is the reason that we do not apply purification in all situations.

**Resolve root weighting.** When using (reach-)resolving<sup>69</sup> for the subgame solve in KLSS in games with no nature actions, the standard algorithm (introduced in Part IV) will choose an opponent information set  $J$  uniformly at random from the set of possible infosets. In reality, the correctness of resolving does not depend on the distribution chosen, so long as it is fully mixed. To be more optimistic, we therefore use a different distribution. We choose an infoset  $J$  via an even mixture of a uniformly random distribution and the distribution of infosets generated from the opponent strategy in the previous iteration of subgame solving. That is, the chance probability of selecting information set  $J$  is

$$\frac{1}{2} \left( \frac{\mathbf{x}(J|I)}{\sum_{J'} \mathbf{x}(J'|I)} + \frac{1}{m} \right),$$

where  $\mathbf{x}(J|I)$  is the probability, according to the blueprint strategy profile, of reaching information set  $J$  given our current information  $I$ ,  $m$  is the number of  $\blacktriangledown$ -infosets in the current subgame, the sum is taken over those same infosets. In this manner, more weight during resolving is given to those positions that were found to be likely in the previous iteration.

**Default strategies in regret matching.** RM+ (and its variants) does not specify what strategy should be used in the event that all regrets are zero, for example, on the first iteration. The default in practice among previous authors has been to use the *uniform random* strategy ( $\mathbf{x}^t = (1/n, \dots, 1/n)$ ). However, we find in practice that it is better to use a default strategy that corresponds to a *guess* as to what a good action may be. In particular, an infoset  $I$  is created the first time a node  $h \in I$  is expanded. At that point, Stockfish’s evaluation function defines a *best action*  $a$  at  $h$  (for the player acting at  $h$ ). The default strategy at  $I$  is set to the pure strategy that always plays  $a$ .

## B Swap Regret and Complexity of NFCE

Joint work with Ioannis Anagnostides, Noah Golowich, Gabriele Farina, and Tuomas Sandholm.

In Part III we discussed no-regret algorithms for various notions of regret. The most robust notion of regret possible is *swap regret*, which corresponds to *all functions*. Very recent breakthrough work (Peng and Rubinstein, 2024; Dagan et al., 2024) showed that low swap regret is achievable with an improved runtime bound (in time  $N^{\tilde{O}(1/\epsilon)}$ , where  $N$  is the size of the game), but still much more slowly than linear-swap regret.

The goal of this project will be to show *lower bounds* on minimizing swap regret or indeed on the complexity of computing the corresponding notion of equilibrium, which is the *normal-form correlated equilibrium*. We already have made some progress toward this: we have achieved a nearly-matching lower bound of  $2^{\Omega(\epsilon^{-1/5})}$  on swap regret for extensive-form strategy sets, matching the normal-form lower bound shown by Dagan et al. (2024) and thus demonstrating that no  $\text{poly}(N, 1/\epsilon)$ -round swap-regret minimization algorithm can exist for extensive-form games.

---

<sup>69</sup>For (reach-)maxmargin solving, there is no prior distribution because the adversary picks the distribution.

## B.1 Notation

This subsection uses the notation from Section 11. In addition, for a function  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , we define

$$V(\phi) := \frac{1}{T} \sum_{t=1}^T \pi^t(\mathbf{x}) \langle \mathbf{u}^t, \phi(\mathbf{x}) \rangle.$$

Thus in particular the total utility experienced by the learner is  $V(\text{Id})$ . After  $T$  rounds, the *swap regret* is  $\max_{\phi} V(\phi) - V(\text{Id})$ , where the maximum is taken over all functions  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ .

## B.2 Impossibility of Swap Regret Minimization

Dagan et al. (2024) showed a lower bound in normal-form games with the following properties:

**Theorem B.1** (Dagan et al. 2024). *For any finite action set  $\mathcal{A} = [N]$ , there exists an normal-form adversary running for  $T \gtrsim N$  rounds, with the following properties:*

1. *The adversary is oblivious. In particular, the adversary selects a sequence  $(u^1, \dots, u^T) \sim \mathcal{D}$  where  $\mathcal{D} \in \Delta(\mathcal{A}^T)$ . The utility of the learner at round  $t$  is then simply  $\mathbb{1}\{a^t = u^t\}$ .*
2. *There exists a strategy  $a^* \in \mathcal{A}$  that is never used by the adversary.*
3. *There exists a partition  $\mathcal{A} = \mathcal{A}_1 \sqcup \dots \sqcup \mathcal{A}_D$  where  $D \lesssim \log T$  with the following property. Within each set  $\mathcal{A}_i$ , number the actions  $\mathcal{A}_i = \{a_{i1}, \dots, a_{iN_i}\}$ . For any sequence  $(u^1, \dots, u^T) \in \text{supp } \mathcal{D}$ , the adversary plays actions in  $\mathcal{A}_i$  only in increasing order. That is, if  $u^t = a_{ij}$  and  $u^{t'} = a_{i'j'}$  and  $t \leq t'$ , then  $j \leq j'$ .*
4. *The swap regret of any learner against this adversary is  $\Omega(D^{-5})$ .*

We have generalized this result to extensive-form games. In particular, we have the following result.

**Theorem B.2.** *There exist arbitrarily large extensive-form strategy sets of dimension  $m$  such that there exists an oblivious adversary which limits any learner to swap regret  $\Omega(\log^{-5} T)$  after  $T = e^{\Theta(m^{1/12})}$  iterations. Thus, there is no online learning algorithm for extensive-form games whose swap regret has the form  $\text{poly}(m)/T^{\Theta(1)}$ .*

We now prove Theorem B.2.

Consider the following family of extensive-form strategy sets, parameterized by natural numbers  $D$  and  $n$ . First the learner picks an index  $i \in [D]$ . Then the environment picks  $j \in [n]$ , and finally the learner picks a binary action. A strategy is identified (up to linear transformations) by a  $\mathbf{x} \in \mathbb{R}^{D \times n}$  where  $\mathbf{x}[i, \cdot] \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n$  and  $\mathbf{x}[i', \cdot] = \mathbf{0}$  if  $i' \neq i$  (i.e.,  $\mathbf{x}$  as a matrix has exactly one nonzero row). For convenience we will use  $\mathcal{X}_i \subset \mathcal{X}$  to denote the set of pure strategies where the learner plays  $i$  at the root. Let  $C$  be the absolute constant required to make the bounds in Theorem B.1 true.

The adversary works as follows. First, for each  $i \in [D]$ , it populates  $\mathcal{A}_i$  with uniformly randomly chosen strategies  $\mathbf{a}_{i1}, \dots, \mathbf{a}_{iN_i} \in \mathcal{X}_i$ . Then the adversary plays as in Theorem B.1.

We will claim that, for any learner against this adversary, there exists a learner against the adversary of Theorem B.1 that achieves a similar swap regret—and thus the swap regret of the former learner must be large. First, we will construct the latter adversary.

Let  $\pi^1, \dots, \pi^T \in \Delta(\mathcal{X})$  be the sequence of distributions played by the learner. Note that  $\pi^t$  can depend on the utilities  $\mathbf{u}^{1:t-1} \in \mathcal{A}$  that are played by the adversary. Consider the sequence  $\bar{\pi}^1, \dots, \bar{\pi}^T \in \Delta(\mathcal{A})$ , where  $\bar{\pi}^t$  is the distribution that samples  $\mathbf{x} \sim \pi^t$  and plays according to  $p_{\mathbf{x}} \in \Delta(\mathcal{A})$ , defined as follows. Let  $\mathbf{x} \in \mathcal{X}_i$  be any strategy. Let  $\epsilon$  be a parameter to be selected later. There are two cases.

1.  $\langle \mathbf{x}, \mathbf{a}_{ij} \rangle \leq \epsilon$  for every  $\mathbf{a}_{ij} \in \mathcal{A}_i$ . Then define  $p_{\mathbf{x}} = \mathbf{a}^*$  deterministically.
2.  $\langle \mathbf{x}, \mathbf{a}_{ij} \rangle > \epsilon$  for some  $\mathbf{a}_{ij} \in \mathcal{A}_i$ . Let  $j$  be the *largest* such index, let  $\beta = \langle \mathbf{x}, \mathbf{a}_{ij} \rangle$ , and define  $p_{\mathbf{x}}$  as the distribution that is  $\mathbf{a}^*$  with probability  $1 - \beta$  and  $\mathbf{a}_{ij}$  with probability  $\beta$ .

A critical property for us will be that the learner cannot “guess in advance” what future unobserved  $\mathbf{a}_{ij}$ s will be, since these are sampled uniformly at random. That is, in Case 2,  $\mathbf{x}$  can only be played with large probability once the adversary has played  $\mathbf{a}_{ij}$ .

To be more formal, we first define some notation. For every  $\mathbf{a}_{ij} \in \mathcal{A}_i$  let  $t_{ij}$  be the first iteration on which the adversary plays  $\mathbf{a}_{ij}$  (or  $t_{ij} = T$  if this never happens). For  $\mathbf{x} \in \mathcal{X}_i$ , if  $\mathbf{x}$  is in Case 1 above then define  $t_{\mathbf{x}} = 0$ , and otherwise define  $t_{\mathbf{x}} = t_{ij}$ , where  $j$  is as in Case 2.

There are two properties that we will critically need to use about  $t_{\mathbf{x}}$ . The first states that the learner cannot place large mass on  $\mathbf{x}$  until after  $t_{\mathbf{x}}$ , because doing so would require the learner to guess a vector heavily correlated with  $\mathbf{a}_{ij}$  before the learner observes  $\mathbf{a}_{ij}$ .

**Lemma B.3.** *Let  $\delta := \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{t_{\mathbf{x}}} \pi^t(\mathbf{x})$ . Then  $\mathbb{E} \delta \leq N e^{-n\epsilon^2/2}$ .*

*Proof.* Since the learner has not yet observed  $\mathbf{a}_{ij}$  at time  $t_{ij}$ , its prior strategy sequence  $\pi^{1:t_{ij}}(\mathbf{x})$  must be independent of  $\mathbf{a}_{ij}$ . Moreover, if  $t \leq t_{\mathbf{x}}$  then there must exist some  $j$  with  $t_{ij} \geq t$  and  $\langle \mathbf{x}, \mathbf{a}_{ij} \rangle \geq \epsilon$ —namely, the  $j$  defining Case 2. Thus we have:

$$\begin{aligned}
\mathbb{E} \delta &= \mathbb{E} \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{t_{\mathbf{x}}} \pi^t(\mathbf{x}) \\
&\leq \mathbb{E} \frac{1}{T} \sum_{i=1}^D \sum_{\mathbf{x} \in \mathcal{X}_i} \sum_{t=1}^T \pi^t(\mathbf{x}) \sum_{j:t_{ij} \geq t} \mathbb{1}\{\langle \mathbf{x}, \mathbf{a}_{ij} \rangle \geq \epsilon\} \\
&= \underbrace{\frac{1}{T} \sum_{i=1}^D \sum_{\mathbf{x} \in \mathcal{X}_i} \sum_{j=1}^{N_i} \mathbb{E} \left[ \sum_{t \leq t_{ij}} \pi^t(\mathbf{x}) \right]}_{\leq N} \underbrace{\mathbb{E} [\mathbb{1}\{\langle \mathbf{x}, \mathbf{a}_{ij} \rangle \geq \epsilon\}]}_{\leq e^{-n\epsilon^2/2}} \leq N e^{-n\epsilon^2/2}.
\end{aligned}$$

where in the last line we use the fact that  $\mathbf{a}_{ij}$  is independent of  $\pi^{1:t_{ij}}(\mathbf{x})$  and then Hoeffding’s inequality.  $\square$

The second property is that, for  $t > t_{\mathbf{x}}$ , utilities of  $\mathbf{x}$  under  $\mathbf{u}^t$  are approximately the same as those of  $p_{\mathbf{x}}$  under the losses in Theorem B.1.

**Lemma B.4.** *For  $t > t_{\mathbf{x}}$ , we have  $\langle \mathbf{x}, \mathbf{u}^t \rangle \leq p_{\mathbf{x}}(\mathbf{u}^t) + \epsilon$ .*

*Proof.* Let  $\mathbf{x} \in \mathcal{X}_i$ . There are two cases. First, if  $\langle \mathbf{x}, \mathbf{a}_{ij} \rangle \leq \epsilon$  for every  $\mathbf{a}_{ij} \in \mathcal{A}_i$ . Then for every  $t$ , we have  $\mathbf{u}^t \notin \text{supp } p_{\mathbf{x}} = \{\mathbf{a}^*\}$  (because the adversary never plays  $\mathbf{a}^*$ ), and  $\langle \mathbf{x}, \mathbf{u}^t \rangle \leq \epsilon$  by definition, so we are done.

Otherwise, let  $j$  be the largest index for which  $\langle \mathbf{x}, \mathbf{a}_{ij} \rangle > \epsilon$ . Then  $t_{\mathbf{x}} = t_{ij}$  by definition, and since  $t > t_{ij}$ , by Property 4 the adversary is no longer allowed to play  $\mathbf{a}_{ij'}$  for  $j' < j$ . Thus, either  $\mathbf{u}^t \notin \text{supp } p_{\mathbf{x}}$  and  $\langle \mathbf{x}, \mathbf{u}^t \rangle \leq \epsilon$ , or  $\mathbf{u}^t = \mathbf{a}_{ij}$ . The former case reduces to the previous paragraph. In the latter case, we have  $\langle \mathbf{x}, \mathbf{u}^t \rangle = \beta = p_{\mathbf{x}}(\mathbf{u}^t)$  by construction of  $f$ .  $\square$

For the rest of this proof we will use  $\bar{V}(\phi)$  to denote the utilities experienced by  $\bar{\pi}^t$  under the utilities in Theorem B.1. That is,

$$\bar{V}(\phi) = \frac{1}{T} \sum_{t=1}^T \sum_{\mathbf{a} \in \mathcal{A}} \bar{\pi}^t(\mathbf{a}) \mathbb{1}\{\phi(\mathbf{a}) = \mathbf{u}^t\} = \frac{1}{T} \sum_{t=1}^T \sum_{\mathbf{x} \in \mathcal{X}} \pi^t(\mathbf{x}) \Pr_{\mathbf{a} \sim p_{\mathbf{x}}} [\phi(\mathbf{a}) = \mathbf{u}^t]$$

By Theorem B.1, there exists a function  $\bar{\phi} : \mathcal{A} \rightarrow \mathcal{A}$  such that<sup>70</sup>  $\mathbb{E}[\bar{V}(\bar{\phi}) - \bar{V}(\text{Id})] \geq 1/CD^5$ . It suffices to show that  $\mathbb{E}[V(\phi) - V(\text{Id})]$  is large. To do this, we will show that, up to small errors,  $V(\text{Id}) \leq \bar{V}(\text{Id})$  and  $V(\phi) \approx \bar{V}(\bar{\phi})$ .

For the first approximation, we have

$$\begin{aligned} V(\text{Id}) &= \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T \pi^t(\mathbf{x}) \langle \mathbf{x}, \mathbf{u}^t \rangle \\ &\leq \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t > t_{\mathbf{x}}} \pi^t(\mathbf{x}) \langle \mathbf{x}, \mathbf{u}^t \rangle + \delta \\ &\leq \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t > t_{\mathbf{x}}} \pi^t(\mathbf{x}) p_{\mathbf{x}}(\mathbf{u}^t) + \epsilon + \delta \\ &\leq \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T \pi^t(\mathbf{x}) p_{\mathbf{x}}(\mathbf{u}^t) + \epsilon + 2\delta = \bar{V}(\text{Id}) + \epsilon + 2\delta. \end{aligned}$$

For the second, we have

$$\begin{aligned} V(\phi) &= \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T \pi^t(\mathbf{x}) \langle \phi(\mathbf{x}), \mathbf{u}^t \rangle \\ &\geq \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t > t_{\mathbf{x}}} \pi^t(\mathbf{x}) \mathbb{E}_{\mathbf{a} \sim p_{\mathbf{x}}} \langle \bar{\phi}(\mathbf{a}), \mathbf{u}^t \rangle - \delta \\ &\geq \frac{1}{T} \sum_{\mathbf{x} \in \mathcal{X}} \sum_{t > t_{\mathbf{x}}} \pi^t(\mathbf{x}) \mathbb{E}_{\mathbf{a} \sim p_{\mathbf{x}}} \langle \bar{\phi}(\mathbf{a}), \mathbf{u}^t \rangle - \epsilon - \delta \\ &\geq \sum_{\mathbf{x} \in \mathcal{X}} \frac{1}{T} \sum_{t=1}^T \pi^t(\mathbf{x}) \mathbb{E}_{\mathbf{a} \sim p_{\mathbf{x}}} \langle \bar{\phi}(\mathbf{a}), \mathbf{u}^t \rangle - \epsilon - 2\delta = \bar{V}(\bar{\phi}) - \epsilon - 2\delta. \end{aligned}$$

Thus,

$$\mathbb{E}[V(\phi) - V(\text{Id})] \geq \mathbb{E}[\bar{V}(\bar{\phi}) - \bar{V}(\text{Id}) - 2\epsilon - 4\delta] \geq \frac{1}{CD^5} - 2\epsilon - 4\delta \geq \frac{1}{3CD^5}$$

by taking

$$\epsilon = \frac{1}{3CD^5} \quad \text{and} \quad n = \frac{2 \log 12NCD^5}{\epsilon^2} = \tilde{O}(D^{11}).$$

and the proof is complete.

### B.3 Polynomial-time expected fixed points

**Definition B.5** (Expected fixed point problem). Let  $\mathcal{X} \subset \mathbb{R}^n$ . The `OracleEFP` problem is the following. Given oracle access to a function  $\phi : \mathcal{X} \rightarrow \text{co } \mathcal{X}$ , computes a distribution  $\pi \in \Delta(\mathcal{X})$  such that

$$\mathbb{E}_{\mathbf{x} \sim \pi} [\phi(\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \pi} [\mathbf{x}].$$

<sup>70</sup>Technically  $\phi$  is a random variable dependent on  $\mathbf{u}^1, \dots, \mathbf{u}^T$ .

### B.3.1 Ellipsoid against hope

In this section we give an overview of the *ellipsoid against hope* (EAH) algorithm (Papadimitriou and Roughgarden, 2005; Jiang and Leyton-Brown, 2011) using the cleaner exposition of Farina and Pipis (2024). Let  $\mathcal{Z} \subset \mathbb{R}^m, \mathcal{Y} \subset \mathbb{R}^n$  be convex, compact sets. At its core, EAH finds feasible points  $\mathbf{z}$  in constrained optimization problems of the form

$$\text{find } \mathbf{z} \in \mathcal{Z} \quad \text{s.t.} \quad \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}^\top \mathbf{A} \mathbf{z} \geq 0, \quad (17)$$

in polynomial time, given two oracles:

1. a “good-enough-response” (GER) oracle<sup>71</sup> that, given  $\mathbf{y} \in \mathcal{Y}$ , outputs a pair  $(\mathbf{z}, \mathbf{A} \mathbf{z}) \in \mathcal{Z} \times \mathbb{R}^d$  such that  $\mathbf{y}^\top \mathbf{A} \mathbf{z} \geq 0$ ; and
2. a separation oracle  $\text{SEP}_{\mathcal{Y}}$  over  $\mathcal{Y}$ .

**Theorem B.6** (Ellipsoid against hope, (Papadimitriou and Roughgarden, 2005; Jiang and Leyton-Brown, 2011; Farina and Pipis, 2024)). *Given a GER oracle and a separation oracle on compact, convex polytope  $\mathcal{Y}$ , there exists a  $\text{poly}(n, L)$ -time algorithm, where  $L$  is the bit complexity of the input,<sup>72</sup> that returns an exact solution  $\mathbf{z}^*$  to (17) that is a mixture of at most  $n$  GER oracle responses.*

### B.3.2 Polynomial-time expected fixed points using ellipsoid against hope

**Theorem B.7.** *Given a linear optimization oracle on  $\mathcal{X}$ , there exists a  $\text{poly}(n, L)$ -time algorithm for the OracleEFP problem, where  $L$  is the bit complexity of the input.<sup>73</sup> The algorithm returns an expected fixed point distribution  $\pi$  supported on at most  $n$  points  $\mathbf{x} \in \mathcal{X}$ .*

*Proof.* The OracleEFP algorithm is equivalent to the following feasibility problem:

$$\text{find } \boldsymbol{\pi} \in \Delta(\mathcal{X}) \quad \text{s.t.} \quad \min_{\mathbf{y} \in [-1, 1]^d} \mathbf{y}^\top \mathbf{A} \boldsymbol{\pi} \geq 0$$

where  $\mathbf{A}$  is the matrix indexed by vectors  $\mathbf{x} \in \mathcal{X}$  whose  $\mathbf{x}$ th column is  $\phi(\mathbf{x}) - \mathbf{x}$ . We apply the EAH algorithm (Theorem B.6). To do this, we need two oracles. The separation oracle on  $[-1, 1]^d$  is trivial. The GER oracle must, given a vector  $\mathbf{y} \in [-1, 1]^d$ , compute a pair  $(\boldsymbol{\pi}, \mathbf{A} \boldsymbol{\pi})$  such that  $\mathbf{y}^\top \mathbf{A} \boldsymbol{\pi} \geq 0$ . Consider  $\boldsymbol{\pi} = \mathbf{e}_{\mathbf{x}}$ , where  $\mathbf{x} = \text{argmin}_{\mathbf{x}} \mathbf{y}^\top \mathbf{x}$ . Such  $\mathbf{x}$  can be computed using the linear optimization oracle. Then  $\mathbf{y}^\top \mathbf{A} \boldsymbol{\pi} = \mathbf{y}^\top (\phi(\mathbf{x}) - \mathbf{x}) \geq 0$  since  $\phi(\mathbf{x}) \in \text{co } \mathcal{X}$ . Thus,  $(\mathbf{e}_{\mathbf{x}}, \phi(\mathbf{x}) - \mathbf{x})$  is a valid GER output.  $\square$

## B.4 Further Research

There are a number of future steps that we intend to take in this project.

1. The most obvious open problem is to show lower bounds on the complexity of computing NFCE in extensive-form games. In particular, our goal is to show the following conjecture.

**Conjecture B.8.** *It is hard (for some natural complexity class, perhaps PPAD) to compute one  $\epsilon$ -NFCE, given a game  $\Gamma$  and parameter  $\epsilon > 0$ .*

2. Theorem B.7 should imply efficient algorithms for  $\Phi$ -equilibria as long as (some superset of) the polytope of deviations  $\Phi$  itself admits an efficient separation oracle. For example, in Section 11 we discussed

<sup>71</sup>Note that the existence of a GER oracle implies, by duality, that (17) is feasible.

<sup>72</sup>Formally,  $L$  bounds 1) the facet complexity of  $\mathcal{Y}$ , and 2) the encoding lengths of all outputs of the two oracles.

<sup>73</sup>Formally,  $L$  bounds the encoding lengths of all outputs of both oracles.

the class of *low-degree deviations*, which can be represented by a reasonable number of constraints; Theorem B.7 should thus imply an efficient algorithm for low-degree correlated equilibria.

The algorithm that achieves this is a “doubly-nested EAH”, where the outer EAH is used to compute the correlated equilibrium itself, and the inner EAH is used as a subroutine to compute expected fixed points. It remains to formalize this argument, especially being wary of any possible issues of numerical precision.

## C Sequential Communication Equilibria

Joint work with Tuomas Sandholm.

One major flaw in computing optimal communication equilibria (as per Part II) is that the resulting equilibria can often contain irrational off-path behavior. In particular, players’ constraints on following recommended actions are only enforced on-path: once the player reaches an info set that cannot be reached in equilibrium, the player will follow any recommended action at that info set, no matter how irrational (even if, for example, it is dominated)

The goal of this project is to develop results analogous to those in Part II for tighter notions of equilibrium. In particular, we focus on the *sequential communication equilibrium* defined by Myerson (1986).

### C.1 Setup

As in Part II, we consider finite extensive-form games  $\Gamma$  with a mediator that is able to communicate with players. Departing from Part II, however, it will be important to us that simultaneous actions be explicitly allowed as part of the model. Thus, we will adopt the notation used in Section 4.2: at every history  $h \in \Gamma$  there is an action set  $A_i(h)$  for each player  $i$  (including chance), and the joint actions  $\mathbf{a} \in \times_{i \in [n] \cup \mathcal{C}} A_i(h)$  are identified with the children of  $h$ . So, every player takes an action at every node. The child of  $h$  identified with action tuple  $\mathbf{a}$  is denoted  $h\mathbf{a}$ . The set of leaves (terminal nodes) of  $\Gamma$  is denoted  $Z$ .

Nonterminal nodes are partitioned into information sets (infosets): for each player,  $\mathcal{I}_i$  is a partition of  $\mathcal{H} \setminus Z$ . The legal action set must be the same if the infosets are the same, *i.e.*,  $A_i(h) = A_i(h') := A_i(I)$  if  $h, h' \in I \in \mathcal{I}_i$ . The collection of infosets for a player  $i$  is  $\mathcal{I}_i$ . For cleanliness of notation we will assume that distinct infosets have disjoint action sets, *i.e.*,  $A_i(I) \neq A_j(J)$  if  $i \neq j$  or  $I \neq J$ .

We will refer to the levels of the game tree as *timesteps*,  $t = 1, \dots, T$  where  $T$  is the depth of the tree and level  $t = 1$  contains only the root. We will also assume that all leaves are at time  $T$ , for simplicity. Assume  $\Gamma$  is timeable, *i.e.*, infosets do not cross different timesteps of the game tree. We use  $\mathcal{H}^t$  to denote the history set at time  $t$ . For notational convenience, let  $A^t := \bigcup_{h \in \mathcal{H}^t, i \in [n]} A_i(h)$  be the set of all actions available at timestep  $t$ , and  $A := \bigcup_{t \in [T]} A^t$  be the set of all actions. We will also use comparators in superscripts, for example,  $A^{>t} = \bigcup_{t' > t} A^{t'}$ .

The *conditional utility*  $u_i(\mathbf{x}|h)$  is the expected utility at the terminal node reached by starting at node  $h$  and following profile  $\mathbf{x}$ .

We will denote the game tree of the mediator-augmented game by  $\hat{\Gamma}$ . In  $\hat{\Gamma}$ , there is an extra player (the *mediator*, female pronouns) who communicates with players. The mediator has no private information of its own except what players report to her. We will generally use hats to distinguish components of  $\hat{\Gamma}$  from components of  $\Gamma$ —*e.g.*,  $\hat{I}_i \in \hat{\mathcal{I}}_i$  is a generic information set,  $\hat{\mathbf{x}}_i \in \hat{\mathcal{X}}_i$  is a generic strategy, and so on. Then, at every time  $t = 1, 2, \dots$ , the following events happen in sequence:

1. each player  $i$  observes his info set  $I_i^t \ni h^t$ , and sends a private message (type report, possibly a lie)  $\hat{I}_i^t \in \mathcal{I}_i$  to the mediator,
2. the mediator sends a private message (action recommendation)  $\hat{a}_i^t \in A_i(\hat{I}_i^t)$  to each player  $i$ .

3. each non-mediator player plays an action  $a_i^t \in A_i(I_i^t)$  (possibly disobeying the action recommendation) in the game  $\Gamma$ . The state advances to  $h^{t+1} := ha^t$ .

The history  $h^t$  eventually reaches a terminal node of  $\Gamma$ , at which point the mediator-augmented game ends and each player gets utility  $u_i(z)$ . The mediator has no utility.

The augmented game  $\hat{\Gamma}$  has three times as many timesteps as  $\Gamma$ . However, in the interest of clean notation, we will only number the timesteps at which the mediator sends recommendations. In particular,  $\hat{\mathcal{H}}^t$  is the set of histories in  $\hat{\Gamma}$  at which the mediator is about to give a recommendation to players at time  $t$  in  $\Gamma$ .

Type reports, of course, could be lies, and players are free to disobey action recommendations.

At a history  $\hat{h} \in \hat{\mathcal{H}}$ , a player  $i$  has been *direct* if the player has so far always sent honest type reports and obeyed all action recommendations. We will overload notation as follows: for each node  $h \in \mathcal{H}$ , we use  $h \in \hat{\mathcal{H}}$  to denote the history in which all players have been direct until  $h$ , and we are in Step 2 above, *i.e.*, it is the mediator's turn to supply action recommendations). A generic strategy for the mediator will be denoted  $\hat{\mu} \in \Xi$ . For  $\hat{h} \in \hat{\mathcal{H}}$  at which the mediator gives recommendations, we will write  $\hat{\mu}(h)_i$  to denote the recommendation given by the mediator to player  $i$ .

We will use  $\hat{\sigma}_i \in \hat{\mathcal{X}}_i$  for  $i \neq 0$  to denote the *direct strategy* of player  $i$ —the direct strategy is the one that always sends honest information reports and obeys honest recommendations.

Let  $B \subseteq A$  be a set of actions. Let  $B^t = B \cap A^t$ . Let  $\hat{\mathcal{X}}_i^B \subseteq \hat{\mathcal{X}}_i$  be the set of player  $i$  strategies in the augmented game (*i.e.*, manipulation plans) in which player  $i$  is direct unless and until he receives a recommendation from set  $B$ .

**Definition C.1.** The set of actions  $B$  is *codominated at time  $t$*  if, for every distribution  $\pi \in \Delta(\mathcal{H}^t \times \Xi)$ , if:

1. with positive probability, the mediator recommends an action in  $B$  to some player at time  $t$ , *i.e.*,

$$\pi(\{(h, \hat{\mu}) \in \mathcal{H}^t \times \Xi : \exists i \text{ s.t. } \hat{\mu}(h)_i \in B\}) > 0, \quad \text{and}$$

2. the mediator never recommends an action in  $B$  after at time  $t$ , *i.e.*,

$$\pi\left(\{(h, \hat{\mu}) \in \mathcal{H}^t \times \Xi : \exists(i, \hat{h}) \in [n] \times \mathcal{H}^{>t} \text{ s.t. } \hat{\mu}(\hat{h})_i \in B\}\right) = 0,$$

then some player  $i$  has a conditionally-profitable deviation that only deviates when recommended an action in  $B$  at time  $t$ , that is, there is a player  $i$  and a strategy  $\hat{x}_i \in \hat{\mathcal{X}}_i^{B^t}$  such that

$$\mathbb{E}_{(h, \hat{\mu}) \sim \pi} [\hat{u}_i(\hat{\mu}, \hat{x}_i, \hat{\sigma}_{-i}|h) - \hat{u}_i(\hat{\mu}, \hat{\sigma}_i, \hat{\sigma}_{-i}|h)] > 0.$$

Set  $B$  is *codominated* if it is codominated at every time  $t$ .

Myerson (1986) showed that the union of codominated sets is codominated, so there is an (inclusion-wise) largest codominated set  $D$ . Call an action *codominated* if it is in  $D$ .

The main result of Myerson (1986) states:

**Theorem C.2** (Theorems 2 and 3 of Myerson (1986)).<sup>74</sup> *Let  $\Gamma_D$  be  $\Gamma$  except with all codominated actions removed in the game. The communication equilibria of  $\Gamma_D$  are precisely the sequential communication equilibria of  $\Gamma$ .*

We will use Theorem C.2 as a *definition* of sequential communication equilibria.

<sup>74</sup>See also the penultimate paragraph of p. 349 of Myerson (1986) for an explicit statement of this form of the result

---

**Algorithm 14:** Computing the set of codominated actions

---

```
1:  $B^1, B^2, \dots, B^T \leftarrow \emptyset$ 
2: for each time  $t = T, \dots, 1$  do
3:   for each action  $a \in A^t$  do
4:     solve program (18) with  $B := B^{>t} \cup \{a\}$ 
5:     if optimal value = 0 then  $B \leftarrow B \cup \{a\}$ 
6: return  $B := \bigcup_{t \in [T]} B^t$ 
```

---

## C.2 Efficient Algorithm for SCE

The main result of this section is the following.

**Theorem C.3** (Main theorem). *There is a polynomial-time algorithm for computing the set  $D$  of all codominated actions.*

*Proof Sketch.* We start by showing that whether a set of actions is codominated at time  $t$  can be decided in polynomial time.<sup>75</sup> Let  $B$  be such a set and let  $a \in B^t$ . Let  $i$  be the player who would play  $a$ , that is,  $i$  is the player for whom  $a \in A_i(h)$  for some  $h$ . Consider the augmented game  $\mathcal{G}_{a,B}$ , defined as follows. At the first timestep, the mediator selects a history  $h^t \in \mathcal{H}^t$ . Only player  $i$  is allowed to deviate, and only when recommended action  $a$ .<sup>76</sup> After timestep  $t$ , all actions from set  $B$  are removed from  $\mathcal{G}$ .

By definition,  $B$  is codominated at  $t$  if  $\mathcal{G}^t$  contains no communication equilibrium in which the mediator recommends  $a$ , without also recommending an action in  $B^{>t}$ . Thus, if  $\Xi_{a,B}^t$  is the polytope of communication equilibria of  $\mathcal{G}_{a,B}$ ,  $B$  is codominated at  $t$  if and only if

$$\max_{\mu \in \Xi_{a,B}^t} \sum_{\substack{h \in \mathcal{H}^t, i \in [n], \\ \mathbf{a} \in A(h): a_i = a}} \hat{\mu}[h\mathbf{a}] = 0. \quad (18)$$

Zhang and Sandholm (2022a) showed that  $\Xi_*^t$  is a polytope that can be efficiently expressed with a polynomial number of variables and constraints. Therefore, the value of the above program is computable in polynomial time.

Now, notice that set  $B$  is codominated if and only if, for all times  $t$  and all actions  $a \in B^t$ , the set  $\{a\} \cup B^{>t}$  is codominated at time  $t$ . This claim follows directly from Definition C.1, by observing that the definition of “codominated at time  $t$ ” does not include any interactions between different actions at time  $t$ : set  $B$  is codominated if, whenever *any* action in  $B^t$  is recommended, some action must be recommended after time  $t$  to counter it. Thus, Algorithm 14 correctly returns the set of all codominated actions, and the proof is complete.  $\square$

**Corollary C.4.** *There is a polynomial-time algorithm for computing an optimal sequential communication equilibrium with respect to any linear objective  $u_M : \mathcal{Z} \rightarrow \mathbb{R}$ .*

<sup>75</sup>This step echoes the proof of Theorem 4 of Myerson (1986), which essentially characterizes the dual of (18).

<sup>76</sup>This restriction can be achieved by limiting the strategy sets of the players in  $\mathcal{G}[a]$ .



## C.3 Remaining Questions and Tasks

1. Flesh out the above proof sketch.
2. Experiments, maybe? (Not sure if meaningful—the algorithm will be *very* slow...)
3. (Perhaps most interesting) What happens for other solution concepts? Myerson’s equivalence proof does not obviously extend to e.g. EFCE, certification eq., etc. Is there a similar characterization?

# D Applications of Generalized Mechanism Design

Joint work with Tuomas Sandholm.

This project is more speculative. I will work on it only if time permits and if we find a sufficiently interesting application, and it may not appear in the final thesis.

We believe that the framework introduced in Part II can be scaled further than exhibited so far, and we intend to demonstrate this scalability by applying it to large problems of economic interest. Below we will propose one possible avenue of research. Deng et al. (2021) study the problem of *dynamic mechanism design with budget constraints*. We first give an overview of their work.

## D.1 Setup

There is a principal (mechanism designer) and an agent.<sup>77</sup> At each timestep  $t = 1, \dots, T$ , the following events happen, in order.

1. The agent observes its type  $\theta_t \sim F_t$  where  $F_t \in \mathcal{F}_t \subseteq \Delta(\Theta_t)$  is the type distribution of the buyer.
2. The agent reports a type  $\hat{\theta}_t \in \Theta_t$  (possibly a lie).
3. The principal observes  $\hat{\theta}_t$  and selects an outcome  $z_t \in \mathcal{Z}_t$ . The agent observes the outcome. The principal and agent get rewards  $u_t^P(z_t)$  and  $u_t^A(\theta_t, z_t)$  respectively.

The above process defines a two-player sequential game. The type distributions  $F_t$  are common knowledge and independent of the reported types or seller allocation/payment selections. A *seller history*  $h_t$  in this game is given by the observations and actions taken by the seller up to and including the disclosed type<sup>78</sup> at time  $t$ . That is,  $h_t = (\hat{\theta}_1, z_1, \dots, \hat{\theta}_t, z_t)$ . Mechanisms<sup>79</sup> are principal strategies in this game. That is, a (deterministic) mechanism is given by a tuple of functions  $(z_t : \mathcal{H}_t \rightarrow \mathcal{Z}_t)_{t=1}^T$ .

<sup>77</sup>Deng et al. (2021) also generalize their results to the multi-agent case, but for simplicity and also to follow their exposition more closely, we will focus here on the single-agent case.

<sup>78</sup>This differs slightly from the notation adopted by Deng et al. (2021), but it is equivalent and slightly cleaner.

<sup>79</sup>Technically these are what Deng et al. (2021) call *clairvoyant mechanisms*, as (implicitly, since we stipulated the common knowledge of distributions  $F_t$ ) these mechanisms can depend on the whole sequence of priors  $F_1, \dots, F_T$ . Deng et al. (2021) also study *non-clairvoyant* mechanisms, which are mechanisms that cannot depend on future type distributions—but we will not discuss these in our formulation for simplicity.

**Incentive compatibility.** We assume the revelation principle and define a mechanism to be *dynamic incentive-compatible* (DIC) if the agent is incentivized to report honestly ( $\hat{\theta}_t = \theta_t$ ) in *every history*<sup>80</sup>  $(\theta_1, \hat{\theta}_1, o_1, \dots, \theta_{t-1}, \hat{\theta}_{t-1}, o_{t-1}, \theta_t)$ .

**Constraints.** The formulation allows *history-dependent constraints* on the mechanism. Formally, let  $\mathcal{Z}_t : \mathcal{H}_t \rightarrow 2^{\mathcal{Z}_t}$  define the constraint at time  $t$ . The mechanism must satisfy  $z_t(h_t) \in \mathcal{Z}_t(h_t)$  for all histories  $h_t$  in order to be valid. The constraints are assumed to be consistent, in the sense that for every  $z_t \in \mathcal{Z}_t(h_t)$  and  $\hat{\theta}_{t+1} \in \Theta_{t+1}$ , we must have  $\mathcal{Z}_t(h_t, z_t, \hat{\theta}_{t+1}) \neq \emptyset$ . (This condition ensures that the mechanism cannot find itself in a state where there is no valid outcome to select.)

**Objective.** Call a mechanism *feasible* if it satisfies DIC and the history-dependent constraints. The objective of the mechanism designer is to select an *optimal* mechanism, that is, a feasible mechanism that maximizes the expected total reward  $\mathbb{E} \sum_{t=1}^T u_t^P(z_t)$  of the principal.

## D.2 Characterization of Optimal Mechanisms

The main result of [Deng et al. \(2021\)](#) concerning (clairvoyant) dynamic mechanism design with such constraints is a characterization of optimal mechanisms in the above setup as *lossless history compression* mechanisms. Informally, the result states that if there is some notion of “state” that is sufficient to define the history-dependent constraints, then that notion is also sufficient to define an optimal mechanism.

More formally, let  $s_t : \mathcal{H}_t \rightarrow \mathcal{S}_t$  define a state  $s_t(h_t)$  corresponding to each history  $h_t$ . Call  $s_t$  a *lossless history compression* (LHC) if

1. constraints depend only on state, *i.e.*, there exists maps  $\tilde{\mathcal{Z}}_t : \mathcal{S}_t \rightarrow 2^{\mathcal{Z}_t}$  with  $\mathcal{Z}_t = \tilde{\mathcal{Z}}_t \circ s_t$  for all  $t$ , and
2. there exist *update rules*  $\Lambda_t : \mathcal{S}_t \times \mathcal{Z}_t \times \Theta_{t+1} \rightarrow \mathcal{S}_{t+1}$  such that  $\Lambda(s_t(h_t), z_t, \hat{\theta}_{t+1}) = s_t(h_t, z_t, \hat{\theta}_{t+1})$ .

Analogously, call a mechanism  $(z_t)_{t=1}^T$  an *LHC mechanism* if there exist functions<sup>81</sup>  $\tilde{z}_t : \mathcal{S}_t \rightarrow \mathcal{Z}_t$  such that  $z_t = \tilde{z}_t \circ s_t$ .

**Theorem D.1** ([Deng et al., 2021](#), Theorem 3.5). *For every feasible mechanism, there exists a feasible LHC mechanism with the same principal objective. In particular, there exists an optimal LHC mechanism.*

## D.3 Example: Auctions with Budget Constraints

The main application of interest to [Deng et al. \(2021\)](#) (and to us in this subsection) is the design of *revenue-maximizing sequential auctions with budget constraints*. In the above language, we have the following components.

- The principal and agent are the buyer and seller, respectively.
- Types  $\Theta_t = [0, 1]$  are valuations.
- Outcomes are  $z_t = (x_t, p_t) \in \mathbb{R}^2$ , where  $x_t \in [0, 1]$  should be interpreted as the amount of item sold to the buyer, and  $p_t \in \mathbb{R}$  is the payment of the buyer. Thus the agent’s utility is  $u_t^A(\theta_t, x_t, p_t) = \theta_t x_t - p_t$ .
- The objective is to maximize revenue. Thus the principal’s utility is  $u_t^P(x_t, p_t) = p_t$ .
- As usual in mechanism design, we insist on *ex-post incentive compatibility* at every timestep. That is, the agent’s total utility must always be nonnegative.

<sup>80</sup>This differs from the formulation we have been using in Part II: the notion in use here is *stronger* because the agent must be incentivized to report honestly *even if it has reported dishonestly in the past*.

<sup>81</sup>The  $\tilde{z}_t$ s can be different for each time  $t$  even if everything else is the same. That is, a “memoryless” LHC mechanism, in which  $|\mathcal{S}_t| = 1$ , may still behave differently in different timesteps.

- The agent has a fixed, common-knowledge *budget*  $B$ .

The last two conditions can be enforced as history-dependent constraints, by setting

$$\mathcal{Z}_t(h_t) = \left\{ z_t = (x_t, p_t) : \sum_{\tau=1}^t u_\tau^A(\hat{\theta}_\tau, z_\tau) \geq 0; \sum_{\tau=1}^t p_\tau \leq B \right\}.$$

An LHC  $s_t$  is given by the map  $s_t : \mathcal{H}_t \rightarrow [0, 1]^3$ , where the three coordinates of  $s_t$  are:

- the current total utility of the buyer,
- the amount of budget the buyer has remaining, and
- the buyer’s current bid.

Formally,  $s_t$  is defined by

$$s_t(\hat{\theta}_1, z_1, \dots, \hat{\theta}_{t-1}, z_{t-1}, \hat{\theta}_t) = \left( \sum_{\tau=1}^t u_\tau^A(\hat{\theta}_\tau, z_\tau), B - \sum_{\tau=1}^t p_\tau, \hat{\theta}_t \right).$$

It is straightforward to formulate the constraints  $\tilde{\mathcal{Z}}_t$  and update rules  $\Lambda_t$  based on this choice of state. Thus, Theorem D.1 guarantees that there exists an optimal mechanism whose decision rule  $\tilde{z}_t$  depends only on  $s_t(h_t)$ .

## D.4 Proposed Research

Recall from Section 7 (Zhang et al., 2023a) that mechanism design settings—including this one—can be formulated as zero-sum games, and indeed we have already used deep RL to compute optimal mechanisms in a four-step repeated auction with budget constraints—a special case of the setup discussed above. That will be our starting point. We propose to explore at least one of the following possible directions for this thesis<sup>82</sup>

**More efficient learning via LHC characterization.** The LHC characterization introduced in the section above can be naturally merged with our deep RL framework in Part II to create a more efficient representation of the optimal policy, because the policy no longer needs to depend on the entire history but only on the three-dimensional state above. We propose to incorporate this improvement into our deep RL model, in the hope of improving the training time or even the overall performance.

**Private budgets.** Deng et al. (2021) explicitly leave the extension to *private budgets* open. In particular, we propose to consider the case where the buyer’s budget  $B$  is known only to the buyer, and the buyer reports (possibly falsely) this budget to the seller before the first round. This case still falls naturally into our model in Part II.

---

<sup>82</sup>This work is still in a very early stage. Indeed, as seen in Table 37, I do not plan to do significant work on this project until after my proposal. Thus these ideas are high-level. I am also open to suggestions from the proposal committee or others regarding other possible applications. Whatever the setting we ultimately choose, the hope is to, at minimum, experimentally validate the framework by demonstrating that it is able to compute good mechanisms in settings where hand analysis does not suffice on its own. The experiments might also yield insights that allow us to develop novel theory for these settings.

**Constraints and communication equilibria.** In the framework of Part II, the natural formulation of individual rationality (IR) is unorthodox, namely by allowing the player the choice to exit the mechanism, returning all allocated items and also any payment (and hence getting utility 0). This formulation is, as discussed in Part II, actually *stronger* than the usual notion of individual rationality, because the mechanism must now also be robust to a buyer who overbids and then chooses to exit if the mechanism forces it to pay higher than its true value. A similar logic applies to other more general constraints, such as budget constraints.

We propose to run an experiment to investigate the effect of choosing between these two formulations of constraints. For example, does it matter for revenue whether IR is formulated in the “traditional” way (*i.e.*, by constraining the outcomes) or in the “exit option” way described above? What about the budget constraint?

**Ex-post vs ex-interim IC.** When there are multiple buyers, there is a clear distinction between *ex-post* and *ex-interim* incentive compatibility: in the former case, the auction must remain incentive-compatible even for a bidder who knows in each stage the bids of the other buyers before choosing its own bid. For single-stage, single-item optimal auction design, this distinction is essentially moot because the auction of Myerson (1981) is the optimal auction in either case. However, to our knowledge, the distinction between ex-post and ex-interim IC in the dynamic setting has not been studied.

Our framework in Part II, once again, captures this distinction naturally. We thus propose to study this distinction using our framework. For example, does enforcing ex-post IC, which is a stronger condition, lead to less revenue? Why or why not?

**Theoretical and practical questions.** In any of the above settings, we can ask both theoretical and practical questions.

- (Practical) What effect does each of the changes have on the optimal revenue? With the experimental setup that we have, answering this should be straightforward: re-run the experiment from Section 7, and compare the results.
- (Theoretical) How would one extend the LHC characterization of optimal mechanisms to these cases? Perhaps the answer to the practical question above would be informed by the results of the practical investigation above.

## References

- Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. *ACM Symposium on Principles of Distributed Computing*, 2006. 49
- Ilan Adler. The equivalence of linear programs and zero-sum games. *Int. J. Game Theory*, 42(1):165–177, 2013. 88
- Matthew Aitchison, Lyndon Benke, and Penny Sweetser. Learning to deceive in multi-agent hidden role games. Stefan Sarkadi, Benjamin Wright, Peta Masters, and Peter McBurney, editors, *Deceptive AI*. Springer International Publishing, 2021. 45
- Ioannis Anagnostides, Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Noah Golowich, and Tuomas Sandholm. Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games. *Symposium on Theory of Computing (STOC)*. ACM, 2021. 89
- Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974. 9, 62, 74, 98
- Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinfeld. Can almost everybody be almost happy? *Conference on Innovations in Theoretical Computer Science (ITCS)*, 2016. 109
- Yu Bai, Chi Jin, Song Mei, Ziang Song, and Tiancheng Yu. Efficient phi-regret minimization in extensive-form games via online mirror descent. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 105
- Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. 7
- Nicola Basilico, Andrea Celli, Giuseppe De Nittis, and Nicola Gatti. Team-maxmin equilibrium: efficiency bounds and algorithms. *AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 8, 57, 59
- Donald Beaver. Multiparty protocols tolerating half faulty processors. *Advances in Cryptology—CRYPTO’89 Proceedings 9*. Springer, 1990. 48
- Dirk Bergemann and Stephen Morris. Bayes correlated equilibrium and the comparison of information structures in games. *Theoretical Economics*, 11(2):487–522, 2016. 129
- Martino Bernasconi, Matteo Castiglioni, Alberto Marchesi, Francesco Trovò, and Nicola Gatti. Constrained phi-equilibria. *International Conference on Machine Learning (ICML)*, 2023. 105
- Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 8:1307–1324, 2007. 96, 97, 105, 112, 119
- Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold’em poker is solved. *Science*, 347(6218), January 2015. 20, 86
- Mark Braverman, Omid Etesami, and Elchanan Mossel. Mafia: A theoretical study of players and coalitions in a partial information environment. *Annals of Applied Probability*, 18:825–846, 2006. 45
- Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. *ACM Conference on Economics and Computation (EC)*, 2018. 131
- Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 12, 20, 132, 133, 135, 137
- Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. 7, 12, 20, 86, 132, 135

- Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. *AAAI Conference on Artificial Intelligence (AAAI)*, 2019a. 18, 20, 55, 101
- Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019b. 7, 12, 20, 71, 132, 135
- Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 132, 133, 138
- Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 132, 133, 138
- Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. *AAAI Conference on Artificial Intelligence (AAAI)*, 2014. 12, 132, 133, 135, 137
- Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019. 19
- Modibo K. Camara, Jason D. Hartline, and Aleck C. Johnsen. Mechanisms for a no-regret agent: Beyond the common prior. *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*. IEEE, 2020. 124
- Colin Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003. 20
- Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1-2): 57–83, 2002. 132
- Luca Carminati, Federico Cacciamani, Marco Ciccone, and Nicola Gatti. A marriage between adversarial team games and 2-player games: Enabling abstractions, no-regret learning, and subgame solving. *International Conference on Machine Learning (ICML)*. PMLR, 2022. 38, 84
- Luca Carminati, Brian Hu Zhang, Federico Cacciamani, Junkang Li, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Efficient representations for team and imperfect-recall equilibrium computation. *in preparation*, 2024a. 30, 60
- Luca Carminati, Brian Hu Zhang, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Hidden-role games: Equilibrium concepts and computation. *ACM Conference on Economics and Computation (EC)*, 2024b. 9, 50, 53, 55, 56, 60
- Andrea Celli and Nicola Gatti. Computational results for extensive-form adversarial team games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 23, 47, 48, 51, 59
- Andrea Celli, Stefano Coniglio, and Nicola Gatti. Private Bayesian persuasion with sequential games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020a. 63, 69
- Andrea Celli, Alberto Marchesi, Gabriele Farina, and Nicola Gatti. No-regret learning dynamics for extensive-form correlated equilibrium. *Advances in Neural Information Processing Systems*, 33:7722–7732, 2020b. 71
- Michal Chalamish and Sarit Kraus. Automed: an automated mediator for multi-issue bilateral negotiations. *Autonomous Agents and Multi-Agent Systems*, 24(3):536–564, 2012. 63
- Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David Juedes, Iyad A Kanj, and Ge Xia. Tight lower bounds for certain parameterized np-hard problems. *Information and Computation*, 201(2):216–231, 2005. 35, 86
- Xinyi Chen, Angelica Chen, Dean Foster, and Elad Hazan. Ai safety by debate via regret minimization, 2023. 97
- Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. *Conference on Learning Theory*, 2012. 17, 86, 89

- Paul F. Christiano. Solving avalon with whispering. 2018. Blog post. Available at <https://sideways-view.com/2018/08/25/solving-avalon-with-whispering/>. 45, 59, 60
- Francis Chu and Joseph Halpern. On the NP-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 2001. 36, 71
- Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021. 98, 101
- Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010. 71, 72
- Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. *Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, 2002. 63
- Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. *ACM Conference on Electronic Commerce (ACM-EC)*, New York, NY, 2004. 63
- Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. From external to swap regret 2.0: An efficient reduction and oblivious adversary for large action spaces. *Symposium on Theory of Computing (STOC)*, 2024. 105, 107, 109, 119, 145, 146
- Christoph Dann, Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Pseudonorm approachability and applications to regret minimization. *International Conference on Algorithmic Learning Theory (ALT)*, 2023. 105
- Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 89
- Yuan Deng, Vahab Mirrokni, and Song Zuo. Non-clairvoyant dynamic mechanism design with budget constraints and beyond. *ACM Conference on Economics and Computation (EC)*, 2021. 94, 153, 154, 155
- Miroslav Dudík and Geoffrey J. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. Jeff A. Bilmes and Andrew Y. Ng, editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. AUAI Press, 2009. 105
- Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. *International Conference on Machine Learning (ICML)*. PMLR, 2019. 10
- Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. 20
- Gabriele Farina and Charilaos Pipis. Polynomial-time linear-swap regret minimization in imperfect-information sequential games. *arXiv preprint arXiv:2307.05448*, 2023. 11, 98, 99, 101, 103, 104, 105, 120
- Gabriele Farina and Charilaos Pipis. Polynomial-time computation of exact phi-equilibria in polyhedral games. *arXiv preprint arXiv:2402.16316*, 2024. 149
- Gabriele Farina and Tuomas Sandholm. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 44, 71, 72, 84, 85
- Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Practical exact algorithm for trembling-hand equilibrium refinements in games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 60

- Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret circuits: Composability of regret minimizers. *International Conference on Machine Learning*, 2019a. [18](#), [101](#)
- Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b. [63](#), [81](#), [87](#), [90](#), [91](#), [104](#)
- Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Efficient regret minimization algorithm for extensive-form correlated equilibrium. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019c. [98](#)
- Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. Coarse correlation in extensive-form games. *AAAI Conference on Artificial Intelligence*, 2020. [9](#), [62](#), [69](#), [74](#), [81](#), [86](#)
- Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. *International Conference on Machine Learning*, 2021a. [18](#), [43](#), [72](#), [73](#), [101](#)
- Gabriele Farina, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *arXiv preprint arXiv:2104.01520*, 2021b. [71](#)
- Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021c. [17](#), [20](#), [43](#), [55](#), [60](#)
- Gabriele Farina, Ioannis Anagnostides, Haipeng Luo, Chung-Wei Lee, Christian Kroer, and Tuomas Sandholm. Near-optimal no-regret learning dynamics for general convex games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. [89](#)
- Francoise Forges. An approach to communication equilibria. *Econometrica: Journal of the Econometric Society*, pages 1375–1385, 1986. [9](#), [49](#), [56](#), [62](#), [64](#), [65](#), [69](#), [70](#), [86](#), [98](#), [122](#), [129](#), [130](#)
- Françoise Forges and Frédéric Koessler. Communication equilibria with partially verifiable types. *Journal of Mathematical Economics*, 41(7):793–811, 2005. [9](#), [62](#), [64](#), [65](#), [69](#), [86](#)
- Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991. [123](#)
- Kaito Fujii. Bayes correlated equilibria and no-regret dynamics. *arXiv preprint arXiv:2304.05005*, 2023. [86](#), [97](#), [105](#), [113](#)
- Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013. [63](#)
- Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. [63](#)
- Anat Ganor and Karthik C. S. Communication complexity of correlated equilibrium with small support. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2018. [96](#)
- Sam Ganzfried and Tuomas Sandholm. Endgame solving in large imperfect-information games. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015a. Early version in AAAI-13 workshop on Computer Poker and Imperfect Information. [20](#), [132](#)
- Sam Ganzfried and Tuomas Sandholm. Safe opponent exploitation. *ACM Transaction on Economics and Computation (TEAC)*, 3(2):8:1–28, 2015b. Best of EC-12 special issue. [136](#)
- Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012. [144](#)



- Oscar Garcia Morchon, Heribert Baldus, Tobias Heer, and Klaus Wehrle. Cooperative security in distributed sensor networks. *2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*, Nov 2007. 45
- Oscar Garcia-Morchon, Dmitriy Kuptsov, Andrei Gurtov, and Klaus Wehrle. Cooperative security in distributed networks. *Computer Communications*, 36(12):1284–1297, 2013. 45
- Angeliki Giannou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Panayotis Mertikopoulos. On the rate of convergence of regularized learning in games: From bandits and uncertainty to optimism and beyond. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021a. 131
- Angeliki Giannou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Panayotis Mertikopoulos. Survival of the strictest: Stable and unstable equilibria under regularized learning with partial information. *Conference on Learning Theory (COLT)*. PMLR, 2021b. 131
- Andrew Gilpin and Tuomas Sandholm. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. *National Conference on Artificial Intelligence (AAAI)*, 2006. 20
- Matthew L. Ginsberg. GIB: Steps toward an expert-level bridge-playing program. *International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, 1999. Morgan Kaufmann. 73
- Paul W. Goldberg and Aaron Roth. Bounds for the query complexity of approximate equilibria. *ACM Trans. Economics and Comput.*, 4(4):24:1–24:25, 2016. 96
- Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. *25<sup>th</sup> international conference on Machine learning*. ACM, 2008. 11, 74, 97, 105, 107, 108, 109, 110, 119
- J Green and J-J Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977. 10, 65, 68
- Amy Greenwald and Keith Hall. Correlated Q-learning. *International Conference on Machine Learning (ICML)*, Washington, DC, USA, 2003. 96, 105
- Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981. 111
- Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2020. 139
- John Harsanyi. Game with incomplete information played by Bayesian players. *Management Science*, 14: 159–182; 320–334; 486–502, 1967–68. 52
- Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000. 17
- Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001. 36
- Elad Hazan and Satyen Kale. Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. *Conference on Neural Information Processing Systems (NeurIPS)*, 2007. 107, 108, 120
- Feng-Hsiung Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002. 7
- Wan Huang and Bernhard von Stengel. Computing an extensive-form correlated equilibrium in polynomial time. *International Workshop on Internet and Network Economics*. Springer, 2008. 62, 71, 105
- Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Asbell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan

- Perez. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024. 45
- Geoffrey Irving, Paul F. Christiano, and Dario Amodei. AI safety via debate. *CoRR*, abs/1805.00899, 2018. 45
- Sergei Izmalkov, Silvio Micali, and Matt Lepinski. Rational secure computation and ideal mechanism design. *Symposium on Foundations of Computer Science (FOCS)*, Washington, DC, 2005. 49
- Eric Jackson. A time and space efficient algorithm for approximately solving large imperfect information games. *AAAI Workshop on Computer Poker and Imperfect Information*, 2014. 132
- Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O’Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. AI alignment: A comprehensive survey. *CoRR*, abs/2310.19852, 2023. 45
- Albert Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *ACM Conference on Electronic Commerce (EC)*, 2011. 149
- Albert Xin Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *Games and Economic Behavior*, 91:347–359, 2015. 71
- Albert Xin Jiang, Ariel Procaccia, Yundi Qian, Nisarg Shah, and Milind Tambe. Defender (mis)coordination in security games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 08 2013. 21
- Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011. 135
- Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011. 9, 63, 69, 120, 129
- Mamoru Kaneko and J Jude Kline. Behavior strategies, mixed strategies and perfect recall. *International Journal of Game Theory*, 24(2):127–145, 1995. 38
- Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. *Theory of Cryptography: Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19–21, 2008. Proceedings 5*. Springer, 2008. 49
- Andrew Kephart and Vincent Conitzer. Complexity of mechanism design with signaling costs. *Autonomous Agents and Multi-Agent Systems*, 2015. 63
- Andrew Kephart and Vincent Conitzer. The revelation principle for mechanism design with signaling costs. *ACM Transaction on Economics and Computation (TEAC)*, 9(1):1–35, 2021. 63
- Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4):528–552, October 1992. 20, 36, 71
- Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. *ACM Symposium on Theory of Computing (STOC)*, 1994. 48, 55, 67
- Yoav Kolumbus and Noam Nisan. Auctions between regret-minimizing agents. Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW ’22: The ACM Web Conference 2022*. ACM, 2022. 124
- Kavya Kopparapu, Edgar A. Duéñez-Guzmán, Jayd Matyas, Alexander Sasha Vezhnevets, John P. Agapiou, Kevin R. McKee, Richard Everett, Janusz Marecki, Joel Z. Leibo, and Thore Graepel. Hidden agenda: a social deduction game with diverse learned equilibria. *CoRR*, abs/2201.01816, 2022. 45
- Vojtěch Kovařík, Dominik Seitz, and Viliam Lisý. Value functions for depth-limited solving in imperfect-information games. *AAAI Reinforcement Learning in Games Workshop*, 2021. 132, 133, 138, 139

- Vojtěch Kovařík, Martin Schmid, Neil Burch, Michael Bowling, and Viliam Lisý. Rethinking formal models of partially observable multiagent decision making. *Artificial Intelligence*, 303:103645, 2022. [26](#)
- Christian Kroer and Tuomas Sandholm. Discretization of continuous action spaces in extensive-form games. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. [88](#)
- Christian Kroer and Tuomas Sandholm. Imperfect-recall abstractions with bounds in games. *ACM Conference on Economics and Computation (EC)*, 2016. [20](#)
- H. W. Kuhn. Extensive games. *Proc. of the National Academy of Sciences*, 36:570–576, 1950a. [20](#)
- H. W. Kuhn. A simplified two-person poker. H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies*, 24, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950b. [43](#), [91](#), [104](#)
- H. W. Kuhn. Extensive games and the problem of information. H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 2 of *Annals of Mathematics Studies*, 28, pages 193–216. Princeton University Press, Princeton, NJ, 1953. [14](#), [107](#)
- Nicolas S. Lambert, Adrian Marple, and Yoav Shoham. On equilibria in games with imperfect recall. *Games and Economic Behavior*, 113:164–185, 2019. [107](#)
- Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. *International Conference on Machine Learning (ICML)*, 2012. [20](#)
- Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. [94](#)
- Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *CoRR*, abs/1908.09453, 2019. [139](#)
- Adam Lerer, Hengyuan Hu, Jakob Foerster, and Noam Brown. Improving policies via search in cooperative partially observable games. *Proceedings of the AAAI conference on artificial intelligence*, 2020. [7](#)
- Yehuda Lindell. Secure multiparty computation (mpc). *Cryptology ePrint Archive*, 2020. [49](#)
- Viliam Lisý, Marc Lanctot, and Michael H Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. *Autonomous Agents and Multi-Agent Systems*, 2015. [43](#), [91](#)
- Heng Liu. Correlation and unmediated cheap talk in repeated games with imperfect monitoring. *International Journal of Game Theory*, 46:1037–1069, 2017. [49](#)
- Weiming Liu, Haobo Fu, Qiang Fu, and Yang Wei. Opponent-limited online search for imperfect information games. *International Conference on Machine Learning (ICML)*. PMLR, 2023. [12](#)
- Hongyao Ma, Fei Fang, and David C Parkes. Spatio-temporal pricing for ridesharing platforms. *Operations Research*, 2021. [63](#)
- Miltiadis Makris and Ludovic Renou. Information design in multistage games. *Theoretical Economics*, 18(4): 1475–1509, 2023. [129](#)
- Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in bayesian games. *Conference on Learning Theory (COLT)*, 2022a. [105](#)
- Yishay Mansour, Alex Slivkins, Vasilis Syrgkanis, and Zhiwei Steven Wu. Bayesian exploration: Incentivizing exploration in bayesian games. *Operations Research*, 70(2):1105–1127, 2022b. [63](#)

- Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game Theory*. Cambridge University Press, 2020. 51
- Stephen McAleer, John B. Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate nash equilibria in large games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 94
- Gatis Midrijanis. Exact quantum query complexity for total boolean functions. *arXiv preprint quant-ph/0403168*, 2004. 113
- James D. Miller, Roman Yampolskiy, Olle Häggström, and Stuart Armstrong. Chess as a testing grounds for the oracle approach to AI safety. Huáscar Espinoza, John A. McDermid, Xiaowei Huang, Mauricio Castillo-Effen, Xin Cynthia Chen, José Hernández-Orallo, Seán Ó hÉigeartaigh, Richard Mallah, and Gabriel Pedroza, editors, *Proceedings of the Workshop on Artificial Intelligence Safety 2021 co-located with the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), Virtual, August, 2021*. CEUR-WS.org, 2021. 45
- Dov Monderer and Moshe Tennenholtz. K-implementation. *J. Artif. Intell. Res.*, 21:37–62, 2004. 124
- Dov Monderer and Moshe Tennenholtz. Strong mediated equilibrium. *Artificial Intelligence*, 173(1):180–195, 2009. 69
- Matej Moravčík, Martin Schmid, Karel Ha, Milan Hladik, and Stephen Gaukrodger. Refining subgames in large imperfect information games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2016. 12, 20, 132, 133, 135, 137
- Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, May 2017. 20, 132, 133, 135
- Dustin Morrill, Ryan D’Orazio, Reza Sarfati, Marc Lanctot, James R. Wright, Amy R. Greenwald, and Michael Bowling. Hindsight and sequential rationality of correlated play. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021a. 105
- Dustin Morrill, Ryan D’Orazio, Marc Lanctot, James R Wright, Michael Bowling, and Amy R Greenwald. Efficient deviation types and learning for hindsight rationality in extensive-form games. *International Conference on Machine Learning (ICML)*. PMLR, 2021b. 105
- Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021. 45
- H. Moulin and J.-P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978. 9, 62, 69, 74, 86, 98
- Roger Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981. 156
- Roger B Myerson. Multistage games with communication. *Econometrica: Journal of the Econometric Society*, pages 323–358, 1986. 9, 49, 56, 62, 64, 69, 86, 98, 122, 129, 130, 150, 151, 152
- John Nash. Equilibrium points in n-person games. *National Academy of Sciences*, 36:48–49, 1950. 15, 47
- Denis Nekipelov, Vasilis Syrgkanis, and Éva Tardos. Econometrics for learning agents. *ACM Conference on Economics and Computation (EC)*, 2015. 124
- Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. *CoRR*, abs/2310.17651, 2023. 105
- Ryan O’Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014. 117
- Aidan O’Gara. Hoodwinked: Deception and cooperation in a text-based game for language models. *CoRR*, abs/2308.01404, 2023. 45

- Christos Papadimitriou and Tim Roughgarden. Computing equilibria in multi-player games. *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Vancouver, BC, Canada, 2005. SIAM. [149](#)
- Christos Papadimitriou, George Pierrakos, Alexandros Psomas, and Aviad Rubinstein. On the complexity of dynamic mechanism design. *Games and Economic Behavior*, 2022. [63](#)
- Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14, 2008. [71](#), [112](#), [120](#)
- Peter S. Park, Simon Goldstein, Aidan O’Gara, Michael Chen, and Dan Hendrycks. AI deception: A survey of examples, risks, and potential solutions. *CoRR*, abs/2308.14752, 2023. [45](#)
- Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. *Symposium on Theory of Computing (STOC)*, 2024. [105](#), [107](#), [109](#), [119](#), [145](#)
- Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022. [20](#)
- Michele Piccione and Ariel Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, pages 3–24, 1997. [107](#)
- Georgios Piliouras, Ryann Sim, and Stratis Skoulakis. Beyond time-average convergence: Near-optimal uncoupled online learning via clairvoyant multiplicative weights update. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. [89](#)
- Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, 1989. [48](#)
- Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. *Conference on Learning Theory*, 2013a. [17](#)
- Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Beyond regret. *Conference on Learning Theory (COLT)*, 2011. [105](#)
- Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 2013b. [17](#), [86](#), [89](#)
- I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962. [16](#), [20](#)
- Sheldon M Ross. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3):621–625, 1971. [91](#)
- Tim Roughgarden. Intrinsic robustness of the price of anarchy. *J. ACM*, 62(5):32:1–32:42, 2015. [95](#)
- Tuomas Sandholm. Abstraction for solving large incomplete-information games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2015a. Senior Member Track. [20](#)
- Tuomas Sandholm. Solving imperfect-information games. *Science*, 347(6218):122–123, 2015b. [20](#)
- Jérémy Scheurer, Mikita Balesni, and Marius Hobbhahn. Technical report: Large language models can strategically deceive their users when put under pressure. *CoRR*, abs/2311.07590, 2023. [45](#)
- Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, G Zacharias Holland, et al. Student of games: A unified learning algorithm for both perfect and imperfect information games. *Science Advances*, 9(46):eadg3256, 2023. [142](#), [143](#), [144](#)
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. [94](#)

- Jack Serrino, Max Kleiman-Weiner, David C Parkes, and Josh Tenenbaum. Finding friend and foe in multi-agent games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 45, 50, 60
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2), 2012. 86
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016. 7, 132, 143
- Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes’ bluff: Opponent modelling in poker. *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005. 43, 91, 104
- Stockfish. <https://stockfishchess.org/>. 132
- Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1-2): 125–159, 2005. 97, 105, 119
- Gilles Stoltz and Gábor Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59(1):187–208, 2007. 96, 105
- Michal Šustr, Vojtěch Kovařík, and Viliam Lisý. Monte Carlo continual resolving for online strategy computation in imperfect information games. *Autonomous Agents and Multi-Agent Systems*, 2019. 132, 133
- Michal Šustr, Vojtěch Kovařík, and Viliam Lisý. Particle value functions in imperfect information games. *AAMAS Adaptive and Learning Agents Workshop*, 2021. 133, 138
- Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 2015. 17, 86, 89
- Oskari Tammelin. Solving large imperfect information games using cfr+. *arXiv preprint arXiv:1407.5042*, 2014. 17, 104, 131
- Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas hold’em. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015. 19, 20, 93
- Emanuel Tewolde, Caspar Oesterheld, Vincent Conitzer, and Paul W. Goldberg. The computational complexity of single-player imperfect-recall games. Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track. 107
- Emanuel Tewolde, Brian Hu Zhang, Caspar Oesterheld, Manolis Zampetakis, Tuomas Sandholm, Paul Goldberg, and Vincent Conitzer. Imperfect-recall games: Equilibrium concepts and their complexity. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.
- Dipty Tripathi, Amit Biswas, Anil Kumar Tripathi, Lalit Kumar Singh, and Amrita Chaturvedi. An integrated approach of designing functionality with security for distributed cyber-physical systems. *J. Supercomput.*, 78(13):14813–14845, sep 2022. 45
- Amparo Urbano and Jose E Vila. Computational complexity and communication: Coordination in two-player games. *Econometrica*, 70(5):1893–1927, 2002. 49
- Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. 94
- Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, Thanasis Lianas, Panayotis Mertikopoulos, and Georgios Piliouras. No-regret learning and mixed nash equilibria: They do not mix. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 131
- Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2): 220–246, 1996. 16, 20, 48

- Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008. [9](#), [36](#), [43](#), [62](#), [65](#), [68](#), [69](#), [70](#), [71](#), [72](#), [74](#), [80](#), [81](#), [84](#), [85](#), [86](#), [98](#), [105](#), [122](#), [129](#)
- Bernhard von Stengel and Daphne Koller. Team-maxmin equilibria. *Games and Economic Behavior*, 21(1): 309–321, 1997. [23](#), [24](#), [47](#), [48](#), [51](#), [56](#)
- Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008. [40](#)
- Kevin Waugh. Abstraction in large extensive games. Master’s thesis, University of Alberta, 2009. [20](#)
- Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024. [112](#)
- Jibang Wu, Zixuan Zhang, Zhe Feng, Zhaoran Wang, Zhuoran Yang, Michael I Jordan, and Haifeng Xu. Sequential information design: Markov persuasion process and its efficient reinforcement learning. *ACM Conference on Economics and Computation (EC)*, 2022. [63](#)
- Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. *CoRR*, abs/2310.18940, 2023. [45](#)
- Brian Hu Zhang and Tuomas Sandholm. Small Nash equilibrium certificates in very large games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020a.
- Brian Hu Zhang and Tuomas Sandholm. Sparsified linear programming for zero-sum equilibrium finding. *International Conference on Machine Learning (ICML)*, 2020b. [135](#)
- Brian Hu Zhang and Tuomas Sandholm. Finding and certifying (near-)optimal strategies in black-box extensive-form games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021a.
- Brian Hu Zhang and Tuomas Sandholm. Subgame solving without common knowledge. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021b. [12](#), [138](#), [142](#)
- Brian Hu Zhang and Tuomas Sandholm. Polynomial-time optimal equilibria with a mediator in extensive-form games. *arXiv preprint arXiv:2206.15395*, 2022a. [9](#), [10](#), [32](#), [49](#), [63](#), [66](#), [67](#), [69](#), [86](#), [87](#), [88](#), [89](#), [129](#), [130](#), [131](#), [152](#)
- Brian Hu Zhang and Tuomas Sandholm. Team correlated equilibria in zero-sum extensive-form games via tree decompositions. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022b. [7](#), [8](#), [38](#)
- Brian Hu Zhang and Tuomas Sandholm. Exponential lower bounds on the double oracle algorithm in zero-sum games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024a.
- Brian Hu Zhang and Tuomas Sandholm. On the outcome equivalence of extensive-form and behavioral correlated equilibria. *AAAI Conference on Artificial Intelligence (AAAI)*, 2024b.
- Brian Hu Zhang, Luca Carminati, Federico Cacciamani, Gabriele Farina, Pierriccardo Olivieri, Nicola Gatti, and Tuomas Sandholm. Subgame solving in adversarial team games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022a.
- Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. *ACM Conference on Economics and Computation (EC)*, 2022b. [10](#), [32](#), [42](#), [43](#), [63](#), [67](#), [91](#), [104](#)
- Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen McAleer, Andreas Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023a. [10](#), [32](#), [49](#), [87](#), [88](#), [89](#), [91](#), [129](#), [130](#), [155](#)

- Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Team belief DAG form: A concise representation for team-correlated game-theoretic decision making. *International Conference on Machine Learning (ICML)*, 2023b. [7](#), [8](#), [10](#), [48](#), [51](#), [56](#), [69](#), [72](#), [81](#), [82](#), [83](#), [84](#), [100](#), [101](#), [108](#), [114](#)
- Brian Hu Zhang, Ioannis Anagnostides, Gabriele Farina, and Tuomas Sandholm. Efficient  $\Phi$ -regret minimization with low-degree swap deviations in extensive-form games. *arXiv preprint arXiv:2402.09670*, 2024a. [11](#), [32](#), [109](#), [110](#), [112](#), [118](#), [119](#)
- Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen Marcus McAleer, Andreas Alexander Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Steering no-regret learners to optimal equilibria. *ACM Conference on Economics and Computation (EC)*, 2024b. [11](#), [120](#), [121](#), [127](#), [130](#), [131](#)
- Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. *under submission*, 2024c. [72](#), [73](#), [81](#), [84](#)
- Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Mediator interpretation and faster learning algorithms for linear correlated equilibria in general sequential games. *International Conference on Learning Representations (ICLR)*, 2024d. [11](#), [32](#), [100](#), [101](#), [104](#), [113](#), [115](#), [116](#)
- Hanrui Zhang and Vincent Conitzer. Automated dynamic mechanism design. *Conference on Neural Information Processing Systems (NeurIPS)*, 34:27785–27797, 2021. [63](#)
- Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Automated mechanism design for classification with partial verification. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. [63](#)
- Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Planning with participation constraints. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022c. [63](#)
- Naifeng Zhang, Stephen McAleer, and Tuomas Sandholm. Faster game solving via hyperparameter schedules. *arXiv preprint arXiv:2404.09097*, 2024e. [20](#)
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593, 2019. [45](#)
- Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. *Conference on Neural Information Processing Systems (NeurIPS)*, 2007. [18](#), [55](#), [86](#), [101](#), [108](#), [113](#), [120](#)