Swap Regret and Correlated Equilibria Beyond Normal-Form Games

ESHWAR RAM ARUNACHALESWARAN, NATALIE COLLINA, YISHAY MANSOUR, MEHRYAR MOHRI, JON SCHNEIDER, AND BALASUBRAMANIAN SIVAN

Swap regret is a notion that has proven itself to be central to the study of general-sum normal-form games, with swap-regret minimization leading to convergence to the set of correlated equilibria and guaranteeing non-manipulability against a self-interested opponent. However, the situation for more general classes of games – such as Bayesian games and extensive-form games – is less clear-cut, with multiple candidate definitions for swap-regret but no known efficiently minimizable variant of swap regret that implies analogous non-manipulability guarantees.

In this paper, we present a new variant of swap regret for polytope games that we call "profile swap regret", with the property that obtaining sublinear profile swap regret is both necessary and sufficient for any learning algorithm to be non-manipulable by an opponent (resolving an open problem of Mansour et al., 2022). Although we show profile swap regret is NP-hard to compute given a transcript of play, we show it is nonetheless possible to design efficient learning algorithms that guarantee at most $O(\text{poly}(d)\sqrt{T})$ profile swap regret in d-dimensional polytope games. Finally, we explore the correlated equilibrium notion induced by low-profile-swap-regret play, and demonstrate a gap between the set of outcomes that can be implemented by this learning process and the set of outcomes that can be implemented by a third-party mediator (in contrast to the situation in normal-form games).

Manuscript submitted for review to the 26th ACM Conference on Economics & Computation (EC'25).

1 Introduction

The theory of regret minimization is one of our most powerful tools for understanding zero-sum games. Consider, for instance, the statement that two players, choosing their strategies in a repeated zero-sum game by running a no-regret learning algorithm, each asymptotically guarantee that they receive their minimax equilibrium value [Freund and Schapire, 1996]. This simple fact lies at the heart of many theoretical results across computer science and economics, and underlies many of the recent super-human level AI performances in games such as Poker and Go [Brown and Sandholm, 2017, Moravcík et al., 2017, Silver et al., 2016].

Many important games are not zero-sum, but instead *general-sum*, allowing for potential alignment between the incentives of the different players. Regret minimization is also a powerful tool for analyzing general-sum games, albeit with some additional caveats. Minimizing the standard notion of *external regret* – the gap between one's utility and the utility of the best fixed action in hindsight – has markedly weaker guarantees than in the zero-sum setting (this only allows convergence to the considerably weaker class of coarse-correlated equilibria, and is vulnerable to certain manipulations by a strategic opponent [Deng et al., 2019]). In these cases, one can get stronger game-theoretic guarantees by minimizing *swap regret* – the gap between one's utility and the counterfactual utility one could have received had they applied the best static *swap function* to all of their actions (e.g., playing rock everywhere they might have previously played scissors). It is a fundamental result in online learning [Blum and Mansour, 2007] that there exist efficient swap regret minimization algorithms for playing in general-sum, normal-form games.

In addition to being general-sum, many games come with some additional structure. For example, in Bayesian games (capturing settings like auctions and markets) players have private information they can use to choose their actions, and in extensive-form games (capturing settings like Poker and bargaining) players take multiple actions in sequence. Both of these classes of games fall under the umbrella of *polytope games*[1]: (two-player[2]) games where each player takes actions in some convex polytope, and where the utilities are provided as bilinear functions of these two action vectors.

Ideally, we would like to apply the theory of regret minimization by minimizing swap regret in the class of polytope games, thereby taking advantage of the previously-mentioned guarantees. It is here where we hit a stumbling block – for reasons we will discuss shortly, there is no single definition of swap regret in polytope games, but instead a wealth of different definitions: linear swap regret, polytope swap regret, normal-form swap regret, Φ-regret, low-degree swap regret, etc. [Fujii, 2023, Gordon et al., 2008, Mansour et al., 2022, Zhang et al., 2024]. Of these notions, the strongest ones which do provide some of the above guarantees (e.g., normal-form swap regret) are conjectured to be intractable to minimize, and the ones for which we have efficient learning algorithms (e.g., linear-swap regret) come with unclear game-theoretic guarantees.

1.1 Main Results and Techniques

Our main contribution in this paper is the introduction of a new variant of swap regret in polytope games which we call *profile swap regret*. We argue that profile swap regret is a particularly natural notion of swap regret in polytope games for two primary reasons:

[1] These games (or slight variants thereof) also appear in the literature under the names *convex games* [Daskalakis et al., 2024a] and *polyhedral games* [Farina et al., 2022]. We choose the term *polytope games* to be consistent with [Mansour et al., 2022], whose work we most directly build off of.

[2] Throughout this paper, we will restrict our attention to games with two players.

- First, many of the utility-theoretic properties that swap regret possesses in general-sum normal-form games extend naturally to profile swap regret in polytope games. Take as one example the property of *non-manipulability*. In [Deng et al., 2019], the authors noticed that common no-external-regret algorithms (such as multiplicative weights) have the property that a strategic opponent can manipulate the play of these learning algorithms to achieve asymptotically more utility than they would be able to achieve against a rational follower (i.e., the "Stackelberg value" of the game). In contrast, no-swap-regret algorithms cannot be manipulated in this way, and it is now understood that the property of incurring sublinear swap regret exactly characterizes when learning algorithms are non-manipulable in general-sum normal-form games [Deng et al., 2019, Mansour et al., 2022].

  Similarly, we show that the property of incurring sublinear profile swap regret is equivalent to the property of being non-manipulable in polytope games (Theorem 10), resolving (one interpretation of) an open question of [Mansour et al., 2022]. We additionally show that no-profile-swap-regret algorithms have minimal asymptotic menus and are Pareto-optimal, two properties established by [Arunachaleswaran et al., 2024] for no-swap-regret algorithms in normal-form games (Theorems 11 and 12).

- Secondly, by extending a recent technique introduced in [Daskalakis et al., 2024a], we show that it is possible to design efficient learning algorithms that guarantee sublinear profile swap regret. That is, we show it is possible to construct explicit learning algorithms that run in time polynomial in the size of the game and number of rounds which incur at most $O(\sqrt{T})$ profile swap regret (Theorem 20). Perhaps more meaningfully, these algorithms guarantee that any opponent can gain at most $O(d\sqrt{T})$ utility over their Stackelberg value in a d-dimensional polytope game by attempting to manipulate this learning algorithm, and are the first known efficient algorithms that provide this guarantee[3].

Before proceeding to discuss these results (and others) in more detail, it is helpful to provide a definition and some intuition for profile swap regret. Consider a game where one player (the "learner", running a learning algorithm) picks actions from a convex polytope $\mathcal{X}$, the other player (the "optimizer", playing strategically) picks actions from a convex polytope $\mathcal{Y}$, and the utility the learner receives from playing action $x \in \mathcal{X}$ against the action $y \in \mathcal{Y}$ is given by $u_L(x, y)$, where $u_L : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is some bilinear function of both players' actions. Consider also a hypothetical transcript of a repeated instance of this game, where the learner has played the sequence of actions $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and the optimizer has played the sequence of actions $\mathbf{y} = (y_1, y_2, \dots, y_T)$. The standard (external) regret of the learner is simply the gap between the utility $\sum_t u_L(x_t, y_t)$ that they received and the best possible utility $\sum_t u_L(x^*, y_t)$ they could have received in hindsight by committing to a fixed action $x^* \in \mathcal{X}$.

How should we define the swap regret of the learner on this transcript? If this were a normal-form game where the learner had m actions, we could define swap regret by comparing the learner's utility to the best possible utility they could obtain by applying a "swap function" $\pi : [m] \to [m]$ to their actions, i.e., $\sum_t u_L(\pi(x_t), y_t)$. Implicit in this definition is the fact that we can extend the swap function π from the set of pure strategies ($[m]$) to the set of mixed strategies (the simplex Δ_m). In the case of normal-form games, there is a clear way to do this: decompose each mixed action as

[3] All other algorithms providing this guarantee do so by minimizing stronger notions of swap regret such as normal-form swap regret, which are conjectured to be hard to minimize (both information-theoretically and computationally, see [Daskalakis et al., 2024b]).

a combination of pure strategies and apply the swap function to each of the pure strategies in this decomposition.

In general polytope games, however, there may not be a unique way to decompose mixed strategies of the learner (elements of $\mathcal{X}$) into pure strategies (vertices of $\mathcal{X}$). For example, if $\mathcal{X} = [0,1]^2$, it is possible to decompose the mixed action $x = (1/2, 1/2)$ as either $\frac{1}{2}(0,0) + \frac{1}{2}(1,1)$, or as $\frac{1}{2}(0,1) + \frac{1}{2}(1,0)$, and these two decompositions could get sent to very different mixed strategies under an arbitrary swap function mapping pure strategies to pure strategies. This has led to a number of different definitions for swap regret in polytope games, including linear swap regret (where we restrict the swap functions to be given by linear transformations, so that they can act directly on mixed actions), polytope swap regret (where we pick the best possible decomposition for the learner in each round), and normal-form swap regret (where we require the learner to directly play a distribution over pure strategies).

Profile swap regret addresses the issue of non-unique convex decompositions in the following way. Given a transcript $(\mathbf{x}, \mathbf{y})$, we construct its corresponding *correlated strategy profile (CSP)* ϕ, which we define to equal the element $\phi = \frac{1}{T}\sum_t x_t \otimes y_t \in \mathcal{X} \otimes \mathcal{Y}$. Note that in normal-form games, the CSP ϕ captures the correlated distribution over pairs of pure strategies played by both players – it plays a somewhat similar role here, allowing us to evaluate the average of any bilinear utility function over the course of play. Now, for any decomposition of ϕ into a convex combination $\phi = \sum_k \lambda_k (x_{(k)} \otimes y_{(k)})$ of product strategy profiles (rank one elements of $\mathcal{X} \otimes \mathcal{Y}$), we define the profile swap regret of this decomposition to equal the maximal increase in utility by swapping each $x_{(k)}$ to the best response to $y_{(k)}$. Finally, we define the overall profile swap regret of this transcript to be the minimum profile swap regret of any valid decomposition of this form.

This definition, although perhaps a little peculiar, has a number of interesting properties:

Utility-theoretic properties. One important consequence of the above definition is that profile swap regret can be computed entirely as a function of the CSP ϕ (unlike stronger regret notions like polytope swap regret and normal-form swap regret). This allows us to build off the work of [Arunachaleswaran et al., 2024], who develop techniques for understanding the subset of possible CSPs an adversary can asymptotically induce against a specific learning algorithm (they call this set the *asymptotic menu* of the learning algorithm). By defining profile swap regret in this way, straightforward generalizations of these menu-based techniques to polytope games suffice to establish the properties of non-manipulability, minimality, and Pareto-optimality for no-profile-swap-regret algorithms.

Efficient no-profile-swap-regret algorithms. Another benefit of of the above definition is that the CSP ϕ is a fairly low-dimensional vector, living in a $(\dim(\mathcal{X}) \cdot \dim(\mathcal{Y}))$-dimensional vector space (contrast this with the amount of information required to store a distribution over the potentially exponential number of vertices of $\mathcal{X}$). This allows us to design learning algorithms that incur at most $O(\sqrt{T})$ profile swap regret by using Blackwell's approachability theorem to force the CSP of the transcript of the game to quickly approach the subset of CSPs with zero profile swap regret.

Interestingly, the computational properties of profile swap regret (and these associated learning algorithms) are somewhat subtle. We prove that it is NP-hard to compute the profile swap regret incurred by a learner during a specific transcript of play $(\mathbf{x}, \mathbf{y})$, even in the special case of Bayesian games (Theorem 16). Ordinarily, this would preclude running the previous approachability-based algorithms efficiently. However, by extending a technique recently introduced in [Daskalakis et al., 2024a] – optimization via semi-separation oracles – we show that it is still possible to implement a variant of the Blackwell approachability algorithm in polynomial time.

Polytope swap regret and game-agnostic learning. In [Mansour et al., 2022], the authors introduced *polytope swap regret* as a measure of swap regret in polytope games that guarantees non-manipulability when minimized, and asked whether any non-manipulable algorithm must necessarily minimize swap regret. Following this, [Rubinstein and Zhao, 2024] answered this question affirmatively for the case of Bayesian games. However, this answer has a minor subtlety – in order to construct Bayesian games where a particular high polytope swap regret algorithm is manipulable, their construction might use games where the optimizer has far more actions available to them than in the games where the learner incurs high regret. They further pointed out that this subtlety is in some sense unavoidable – if you fix the number of actions and types of the two players, there are learning algorithms which incur high polytope swap regret but that are not manipulable.

We study this phenomenon in general polytope games by drawing a distinction between *game-aware learning algorithms* – algorithms that can see the sequence of actions $\mathrm{y} \in \mathcal{Y}$ the optimizer is directly playing – and *game-agnostic learning algorithms* – algorithms that can only see the sequence of induced counterfactual rewards (i.e., the function sending an action $x \in \mathcal{X}$ to the utility $u(x, y_t)$ they would have received if they played x in round t). While profile swap regret characterizes non-manipulability for game-aware algorithms, we show that polytope swap regret characterizes non-manipulability for game-agnostic algorithms (Theorem 23), thus providing an analogue of the result of [Rubinstein and Zhao, 2024] for general polytope games.

Implications for equilibrium computation. We finally turn our attention to the question of how to define and compute correlated equilibria (CE) in polytope games. We argue that there are two main motivations for studying correlated equilibria, which are often conflated due to their agreement in the case of normal-form games. The first is the idea that a correlated equilibrium is an outcome that is inducible by a third party mediator providing correlated recommendations to all players. This idea is appealing from a mechanism design point-of-view, as one can imagine directly implementing the correlated equilibrium of our choice in a game by constructing such a mediator (e.g., installing a traffic light).

However, in many settings of interest, there is no explicit mediator. A second motivation for studying CE is that a correlated equilibrium represents a possible outcome of repeated strategic play between rational agents. We can still relate this back to our original mediator motivation by saying that an outcome is a correlated equilibrium if every player can individually imagine a “one-sided” mediator protocol incentivizing this scheme, where only that player’s incentive constraints need to be met (this is in contrast to the “two-sided” mediator protocol above, which must work for all players simultaneously).

We show that, in polytope games, this first form of CE corresponds to *normal-form CE* (reached by normal-form swap regret dynamics) whereas the second form of CE corresponds to *profile CE* (reached by profile swap regret dynamics). We further show that there is a gap between these two notions of correlated equilibria in general polytope games, exhibiting a separation which does not appear in the case of normal-form games. Finally, we note that our efficient no-profile-swap-regret algorithms mentioned above allow us to compute a profile CE in general games without being able to optimize over the set of profile CE, echoing the results of [Papadimitriou and Roughgarden, 2008] for computing correlated equilibria in succinct multiplayer games.

Additional Related Work. We discuss additional related work in Appendix C.

2 Model and Preliminaries

Notation. Given two convex sets $C_1 \subseteq \mathbb{R}^{d_1}$ and $C_2 \subseteq \mathbb{R}^{d_2}$, we define their tensor product $C_1 \otimes C_2$ to be the subset of $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \simeq \mathbb{R}^{d_1 d_2}$ equal to the convex hull of all vectors of the form $c_1 \otimes c_2 = c_1 c_2^\top$ for $c_1 \in C_1$ and $c_2 \in C_2$. We write $\mathcal{B}_d(R)$ to denote the d-dimensional ball of radius R centered at the origin, and $\mathcal{B}_d(x, R)$ to denote the d-dimensional ball of radius R centered at x (sometimes omitting d clear from context).

Unless otherwise specified, all norms $||\cdot||$ refer to the ℓ_2 norm in the ambient space. Note that any bilinear function $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ corresponds to a vector $\hat{f}$ such that $f(x, y) = \langle \hat{f}, x \otimes y \rangle$; we define the various norms of f (e.g. $||f||_1$, $||f||_2$) to equal those of $\hat{f}$.

2.1 Polytope Games

We begin by introducing the notion of an *polytope game*: a two-player game where the action sets of both players are convex polytopes[4] and where the payoffs of both players are provided by bilinear functions over these two sets. Formally, a polytope game G is a game between two players; we call these two players the *learner* and the *optimizer*. The learner selects an action x from the d_L-dimensional bounded convex polytope $\mathcal{X} \subset \mathcal{B}_{d_L}(1)$ and the optimizer selects an action y from the bounded convex polytope $\mathcal{Y} \subset \mathcal{B}_{d_O}(1)$ (note that $\mathcal{X}$ and $\mathcal{Y}$ can have different dimensions). After doing so, the learner receives utility $u_L(x, y)$ and the optimizer receives utility $u_O(x, y)$, where u_L and u_O are both bounded bilinear functions satisfying $||u_L||_1, ||u_O||_1 \le 1$ (so they map $\mathcal{X} \times \mathcal{Y}$ to $[-1, 1]$). These two action sets ($\mathcal{X}$ and $\mathcal{Y}$) and payoff functions (u_L and u_O) define the game G.

We briefly note here that polytope games (for specific choices of $\mathcal{X}$ and $\mathcal{Y}$) capture a variety of different strategic settings. For example:

- **Normal-form games**: When $\mathcal{X} = \Delta_m$ and $\mathcal{Y} = \Delta_n$, this captures the class of normal-form games where the learner has m available actions and the optimizer has n available actions.
- **Bayesian games**: When $\mathcal{X} = (\Delta_m)^{c_L}$ and $\mathcal{Y} = (\Delta_n)^{c_O}$, this captures the class of Bayesian games where again the learner and optimizer have m and n available actions respectively, but in addition the learner is one of c_L types and the optimizer is one of c_O types. Here we should interpret the set $(\Delta_m)^{c_L}$ as representing all functions mapping a learner's type (an element of $[c_L]$) to a learner's mixed action (an element of Δ_m); we can interpret the set $(\Delta_n)^{c_O}$ similarly.
- **Extensive-form games**: A two-player extensive form game can be written as a polytope game by letting $\mathcal{X}$ and $\mathcal{Y}$ be the sequence-form polytopes for the two players. See e.g. [Von Stengel, 1996] for details.

Generally, we will be interested in polytope games that are played repeatedly for T rounds (and even more generally, the limiting behavior of such games as T approaches infinity). In a repeated setting, we will let $x_t \in \mathcal{X}$ denote the action taken by the learner at round t, and let $y_t \in \mathcal{Y}$ denote the action taken by the optimizer at round t. Given a transcript of a repeated game where the learner has played the sequence of actions $x_1, x_2, \dots, x_T$ and the optimizer has played the sequence of actions $y_1, y_2, \dots, y_T$, we say that the *correlated strategy profile (CSP)* ϕ corresponding to this transcript is given by

[4] Almost all the results should extend to the slightly more general setting of arbitrary bounded convex sets. We focus on the specific setting of polytopes as this captures essentially all the primary settings of interest (e.g. Bayesian games and extensive-form games).

$$\phi = \frac{1}{T}(x_1 \otimes y_1 + x_2 \otimes y_2 + \cdots + x_T \otimes y_T) \in \mathcal{X} \otimes \mathcal{Y}. \tag{1}$$

By construction, the CSP ϕ provides sufficient information to evaluate the average value of any bilinear function $f(x_t, y_t)$ over the transcript of play. Under the minor assumption[5] that $\mathcal{X}$ and $\mathcal{Y}$ are contained within proper affine subspaces of $\mathbb{R}^{d_L}$ and $\mathbb{R}^{d_O}$ respectively, the CSP ϕ is also sufficient to evaluate the average of any bi-affine function $f(x_t, y_t)$ over this transcript. We will therefore assume that $\mathcal{X}$ and $\mathcal{Y}$ satisfy this assumption throughout the rest of the paper, and write $f(\phi) = \frac{1}{T}\sum_t f(x_t, y_t)$ for any such bi-affine function f (e.g, $u_L(\phi)$ is the average learner utility over this transcript of play).

Given a polytope game, we will write $\mathsf{BR}_L(x)$ to denote the set of the learner's best responses to the optimizer's action y, i.e., $\mathsf{BR}_L(y) = \{x \in \mathcal{X} \mid u_L(x, y) = \max_{x^* \in \mathcal{X}} u_L(x^*, y)\}$. We say an action $x \in \mathcal{X}$ for the learner is *strictly dominated* if it is not the best response to any action, i.e., there does not exist an $y \in \mathcal{Y}$ such that $x \in \mathsf{BR}_L(y)$. We say an action $x \in \mathcal{X}$ for the learner is *weakly dominated* if it is *not* strictly dominated but it is impossible for the optimizer to uniquely incentivize x, i.e., there does not exist an $y \in \mathcal{Y}$ such that $\mathsf{BR}_L(y) = \{x\}$. We say a polytope game G is *non-degenerate* if none of the vertices of $\mathcal{X}$ (the learner's extremal actions) are weakly dominated.

2.2 Learning Algorithms and Regret

Thus far, we have made no distinction between the role of the optimizer and the learner. The difference, of course, is that the learner will play this repeated game by running a learning algorithm.

A *learning algorithm* $\mathcal{A}$ for the game G is a family $\{\mathcal{A}^T\}_{T \in \mathbb{N}}$ of horizon-dependent learning algorithms. A horizon-dependent learning algorithm $\mathcal{A}^T$ for time horizon T is a collection of T functions $\mathcal{A}^T_1, \mathcal{A}^T_2, \dots, \mathcal{A}^T_T$, where $\mathcal{A}^T_t$ describes the learner's play at time t as a function of the optimizer's play up until round $t-1$, i.e., $\mathcal{A}^T_t(y_1, y_2, \dots, y_{t-1}) = x_t$. Note that as written, a learning algorithm is specific to the given game G, and a learning algorithm for one game G cannot necessarily easily be applied to a separate game G'. The important caveat to this is that a learning algorithm does not depend on the optimizer's utility function u_O, and we expect a robust learning algorithm $\mathcal{A}$ for a game G to perform well against several different opponents with different choices of u_O.

We can measure the performance of a learning algorithm via some version of *regret*. For example, the *external regret* of a learning algorithm is the gap between its utility and the counterfactual utility it would have received if it played the best fixed action in hindsight. Formally, if the learner has played the sequence of actions $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and the optimizer has played the sequence of actions $\mathbf{y} = (y_1, y_2, \dots, y_T)$, the external regret of the learner is given by

$$\mathsf{Reg}(\mathbf{x}, \mathbf{y}) = \max_{x^* \in \mathcal{X}} \sum_{t=1}^{T} u_L(x^*, y_t) - \sum_{t=1}^{T} u_L(x_t, y_t).$$

Note that since u_L is bilinear, we can also compute the external regret directly from the CSP ϕ corresponding to this transcript of play. In particular, we can alternatively write

[5] This assumption is true for the examples mentioned above, and can always be made to be true by augmenting $\mathcal{X}$ and $\mathcal{Y}$ with an additional dummy coordinate fixed to equal 1.

$$\mathsf{Reg}(\phi) = T \cdot \left(\max_{x^* \in \mathcal{X}} u_L(x^* \otimes \mathrm{proj}_{\mathcal{Y}}(\phi)) - u_L(\phi) \right),$$

where $\mathrm{proj}_{\mathcal{Y}}(\phi) = \frac{1}{T}(y_1 + y_2 + \cdots + y_T)$ [6] is the projection of ϕ onto the optimizer's action space (i.e., the average action taken by the optimizer over the course of the game).

In these notes we will primarily care about forms of regret stronger than external regret, namely variants of *swap regret*. For the case of normal-form games (when the learner's action set $\mathcal{X} = \Delta_m$ is just the simplex over m pure actions), the swap regret of a transcript of play can be defined via

$$\mathsf{SwapReg}(\mathbf{x}, \mathbf{y}) = \max_{\pi^* : [m] \to [m]} \sum_{t=1}^{T} u_L(\pi^*(x_t), y_t) - \sum_{t=1}^{T} u_L(x_t, y_t). \tag{2}$$

Intuitively, this should be thought of as the gap between the utility received by the learner and the maximum utility the learner could have received if they applied a specific "swap function" π^* to their sequence of play: a function which transforms each pure strategy $i \in [m]$ of a learner to a new pure strategy $\pi(i) \in [m]$. Implicit in this definition is the fact that although π^* is a function on *pure strategies* (elements of $[m]$), we can uniquely extend π^* to act on *mixed strategies* (elements $x \in \Delta_m$) via $\pi^*(x)_i = \sum_{j | \pi(j) = i} x_j$. That is, the weight of action i in $\pi^*(x)$ is equal to the total weight of actions j in x that map to i under π^*.

When the learner's action set $\mathcal{X}$ is an arbitrary polytope (and not a simplex), it is not clear how to perform this extension from swap functions on "pure strategies" (vertices of $\mathcal{X}$) to swap functions on "mixed strategies" (points within $\mathcal{X}$). For this reason, it is not clear how to directly generalize the notion of swap regret to polytope games, and a couple different plausible definitions have previously been proposed. We present three of these definitions below, in increasing order of strength[7]: *linear swap regret*, *polytope swap regret*, and *normal-form swap regret*.

Linear swap regret. In linear swap regret, we constrain all of our swap functions to be linear transformations, which can then be applied directly to the mixed strategies played by the learner. Specifically, let Ψ be the set of all affine linear transformations that map the set $\mathcal{X}$ to itself. The *linear swap regret* of a transcript with learner actions $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ and optimizer actions $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ is given by

$$\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = \max_{\psi \in \Psi} \sum_{t=1}^{T} u_L(\psi(x_t), y_t) - \sum_{t=1}^{T} u_L(x_t, y_t).$$

As was the case with external regret (and swap regret over the simplex), we can write linear swap regret as a function of the CSP ϕ corresponding to this transcript of play via

$$\mathsf{LinSwapReg}(\phi) = T \cdot \left(\max_{\psi \in \Psi} u_L(\psi(\phi)) - u_L(\phi) \right),$$

[6] This operation is well defined – we can extract the marginal action of the optimizer from the CSP ϕ since it an average of a bi-affine function $g(x_t, y_t) = y_t$ applied to each entry in the transcript.

[7] Meaning that linear swap regret will always be the smallest (and easiest to minimize) and normal-form swap regret will always be the largest (and hardest to minimize).

where we extend ψ to act on CSPs ϕ via $\psi(x \otimes y) = \psi(x) \otimes y$.

Polytope swap regret. Another way to adapt the definition of swap regret to polytope games is to still keep swap functions that act (possibly non-linearly) on the set of pure strategies, but choose the decomposition of the learner's mixed actions into pure actions in the way that is optimal for the learner. This is the approach taken by [Mansour et al., 2022] in the definition of polytope swap regret.

Formally, we define the *polytope swap regret* $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$ of a transcript with learner actions $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and optimizer actions $\mathbf{y} = (y_1, y_2, \dots, y_T)$ as follows.

(1) First, decompose each of the learner's actions x_t into a convex combination of the extreme points $\mathcal{V}(\mathcal{X})$ of $\mathcal{X}$. We will denote this decomposition by $x_t^V \in \Delta(\mathcal{V}(\mathcal{X}))$. Note that there may be many ways to do this decomposition: we are free to choose any of them and will eventually choose the decompositions that *minimize* our eventual regret.

(2) For any swap function $\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$, we will let $\pi(x_t^V) \in \Delta(\mathcal{V}(\mathcal{X}))$ denote the resulting distribution over the extreme points of $\mathcal{X}$ after applying π (i.e., in the same manner as in the definition of swap regret), and $\overline{\pi}(x_t^V)$ be the element of $\mathcal{X}$ formed by taking the average action in $\pi(x_t^V)$.

(3) Finally, we define

$$\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}^V} \max_{\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \left(\sum_{t=1}^{T} u_L(\overline{\pi}(x_t^V), y_t) - u_L(x_t, y_t) \right).$$

That is, after the learner picks their preferred decomposition of actions into x_t^V, the adversary picks the swap function π that maximizes the resulting regret from comparing the original sequence of actions x_t to the transformed sequence of actions $\overline{\pi}(x_t^V)$.

Normal-form swap regret. Finally, we can attempt to directly use the original definition of swap regret by "expanding" any polytope game to a normal-form game where pure strategies for the optimizer and learner correspond to extreme points of $\mathcal{X}$ and $\mathcal{Y}$ (equivalently, forcing the learner to specify their own decomposition of their mixed strategies into pure strategies each round). In particular, we define the *vertex game* corresponding to a polytope game G to be the normal-form game where the learner plays mixtures of actions in $\mathcal{V}(\mathcal{X})$ and the optimizer plays mixtures of actions in $\mathcal{V}(\mathcal{Y})$ (i.e., with $\mathcal{X}' = \Delta(\mathcal{V}(\mathcal{X}))$ and $\mathcal{Y}' = \Delta(\mathcal{V}(\mathcal{Y}))$). If the learner plays a sequence of vertex game actions $\mathbf{x}^V = (x_1^V, x_2^V, \dots, x_T^V)$ and the optimizer plays a sequence of vertex game actions $\mathbf{y}^V = (y_1^V, y_2^V, \dots, y_T^V)$, the *normal-form swap regret* of this transcript of play is given by

$$\mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V) = \max_{\pi^* : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \sum_{t=1}^{T} u_L(\pi^*(x_t^V), y_t^V) - \sum_{t=1}^{T} u_L(x_t^V, y_t^V). \tag{3}$$

Note that this is simply the original definition of swap regret (2) as applied to the vertex game.

Technically, this definition of swap regret is of a different flavor from the previous definitions, in that there is no clear way to take a transcript of play $\mathbf{x} = (x_1, \dots, x_T)$ and $\mathbf{y} = (y_1, \dots, y_T)$ of the original polytope game and evaluate the normal-form swap regret of this transcript. However, note that we can always go the other direction – given a transcript of play $(\mathbf{x}^V, \mathbf{y}^V)$ of the vertex game, we can always construct a corresponding transcript of play $(\mathbf{x}, \mathbf{y})$ for the polytope game (e.g., by letting $y_t = \mathbb{E}[y_t^V] \in \mathcal{Y}$) and evaluate the other measures of swap regret on this transcript.

Comparisons between regret definitions. We conclude this section with a couple of comparisons between regret notions. First, we note that for the case of normal-form games, all three of these notions reduce to the standard notion of swap regret (as we would expect).

Theorem 1. *Fix a polytope game G with $\mathcal{X} = \Delta_m$. Then for any transcript of play $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and $\mathbf{y} = (y_1, y_2, \dots, y_T)$ in G, we have that*

$$\mathsf{SwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{NFSwapReg}(\mathbf{x}, \mathbf{y}).$$

(Note that for normal-form games, the vertex game is identical to the original game, and therefore the quantity $\mathsf{NFSwapReg}(\mathbf{x}, \mathbf{y})$ is well-defined).

The following theorem shows that the above three forms of swap regret are ordered as originally described. Moreover, this ordering is “strict” – minimizing one of these regret notions implies nothing about the larger regret notions.

Theorem 2. *Let $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and $\mathbf{y} = (y_1, y_2, \dots, y_T)$ be a transcript of play for a given polytope game. Let $\mathbf{x}^V = (x_1^V, x_2^V, \dots, x_T^V)$ and $\mathbf{y}^V = (y_1^V, y_2^V, \dots, y_T^V)$ be a transcript of play for the corresponding vertex game such that $x_t = \mathbb{E}[x_t^V]$ and $y_t = \mathbb{E}[y_t^V]$ for each $t \in [T]$. Then:*

$$\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V).$$

Moreover, all these inequalities are asymptotically strict in the following sense: for each two neighboring definitions of swap regret, there exists a family of transcripts for which the larger swap regret grows as $\Omega(T)$ but the smaller swap regret is zero.

We will often be interested in the worst-case regret incurred by some algorithm. For a horizon-dependent learning algorithm $\mathcal{A}^T$, we let $\mathsf{Reg}(\mathcal{A}^T)$ denote the maximum value of $\mathsf{Reg}(\mathbf{x}, \mathbf{y})$ over all transcripts of length T obtainable by playing against $\mathcal{A}^T$. For a general learning algorithm $\mathcal{A}$, we will let $\mathsf{Reg}(\mathcal{A})$ denote the function mapping T to $\mathsf{Reg}(\mathcal{A}^T)$ (so e.g., we may have $\mathsf{Reg}(\mathcal{A}) = O(\sqrt{T})$). We extend this notation to other notions of regret (e.g., SwapReg, LinSwapReg, PolySwapReg, NFSwapReg) in the obvious way.

2.3 Manipulability and Menus

Given the above range of possible definitions for swap regret in polytope games, which ones should we target when learning in games? Of course, we can always try to minimize the strongest form of swap regret above (normal-form swap regret), but this might come with trade-offs in the form of worse regret bounds and increased algorithmic complexity.

Instead, a more principled approach is to understand why we might want to minimize swap regret in normal-form games in the first place, and then pick the variant of swap regret that gives us comparable guarantees for polytope games. In particular, swap-regret minimization has a number of nice game-theoretic consequences in the form of convergence to certain classes of equilibria and robustness of the learner to certain dynamic manipulations of the optimizer. Informally, we can state some of these consequences as follows:

(1) **(Non-manipulability)** No-swap-regret learning algorithms are "non-manipulable", in the sense that if a learner is running a no-swap-regret learning algorithm, the optimizer can do nothing asymptotically better than playing a fixed (possibly mixed) action every round. Moreover, every non-manipulable no-regret learning algorithm must also be no-swap-regret.

(2) **(Minimality)** No-swap-regret learning algorithms form a "minimal core" of all no-regret learning algorithms: any CSP ϕ that an optimizer can implement against a no-swap-regret learning algorithm can be implemented against any no-regret algorithm.

(3) **(Pareto-optimality)** No-swap-regret learning algorithms are Pareto-optimal: there is no learning algorithm that performs asymptotically better than a no-swap-regret learning algorithm against every possible optimizer. Notably, some classic no-regret learning algorithms like Follow-The-Regularized-Leader are *not* Pareto-optimal in this sense.

(4) **(Correlated equilibria)** When both players in the game run no-swap-regret learning algorithms, the time-averaged CSP converges to the set of correlated equilibria of the game.

We will table the discussion of equilibria until Section B – it is complicated by the fact that it also is not exactly clear what the exact definition of correlated equilibria should be for polytope games. Instead, we will focus on the first three points for now, which all have clear utility-theoretic interpretations that we can extend to general polytope games. To more formally define them, it is useful to introduce the concept of menus.

Given a horizon-dependent learning algorithm $\mathcal{A}^T$, we define the *menu* $\mathcal{M}(\mathcal{A}^T)$ of this algorithm to be the convex hull of all CSPs that an optimizer can implement against $\mathcal{A}^T$. That is, $\mathcal{M}(\mathcal{A}^T) \subseteq \mathcal{X} \otimes \mathcal{Y}$ is the convex hull of all points of the form $\frac{1}{T}\sum_{t=1}^{T} x_t \otimes y_t$ where $x_t = \mathcal{A}_t^T(y_1, y_2, \dots, y_{t-1})$, and $y_1, y_2, \dots, y_T$ is an arbitrary sequence of optimizer actions in $\mathcal{Y}$. For a general learning algorithm $\mathcal{A}$, we define the *asymptotic menu* $\mathcal{M}(\mathcal{A})$ to be the limit of the sequence of menus $\mathcal{M}(\mathcal{A}^1), \mathcal{M}(\mathcal{A}^2), \dots$ in the Hausdorff norm[8].

There is a simple test (coming from Blackwell approachability) for whether a convex set $\mathcal{M}$ is the asymptotic menu of some learning algorithm: it suffices that $\mathcal{M}$ contain some CSP of the form $x \otimes y$ for each $y \in \mathcal{Y}$.

Theorem 3. *A convex set $\mathcal{M}$ is the asymptotic menu of some learning algorithm iff $\mathcal{M}$ contains some CSP of the form $x \otimes y$ for each $y \in \mathcal{Y}$.*

Given a polytope game G (with a specific optimizer payoff u_O), we define the *Stackelberg value* $\mathsf{Stack}(G, u_O)$ to be the optimal payoff the optimizer can receive if they commit to playing a fixed strategy $y \in \mathcal{Y}$ and the learner best responds (breaking ties in favor of the optimizer). That is,

$$\mathsf{Stack}(G, u_O) = \max_{y \in \mathcal{Y}} \max_{x \in \mathsf{BR}_L(y)} u_O(x, y).$$

We make the dependence on u_O explicit here because we will often want to consider the effect of changing u_O while keeping all other parameters of the game ($\mathcal{X}$, $\mathcal{Y}$, and u_L) the same – this captures the strategic problem of facing an optimizer with unknown rewards.

[8] Throughout this paper, for simplicity, we will assume that this sequence of convex sets always converges. In general, one can always take some subsequence of this sequence that converges – see Appendix D of [Arunachaleswaran et al., 2024] for details.

Against any no-(external)-regret learning algorithm $\mathcal{A}$, an optimizer can asymptotically achieve $\mathsf{Stack}(G, u_O)$ utility per round by simply playing their optimal Stackelberg action every round. Intuitively, an algorithm of the learner is non-manipulable if the optimizer cannot significantly increase their utility beyond this Stackelberg value by playing a strategy that changes over time.

It is convenient to phrase this concept of non-manipulability in the language of menus. For any menu $\mathcal{M}$, define $V_O(\mathcal{M}, u_O) = \max_{\phi \in \mathcal{M}} u_O(\phi)$ to be the maximum optimizer utility of any CSP ϕ in $\mathcal{M}$. If $\mathcal{M}$ is the menu of an algorithm $\mathcal{A}$, note that this is just the maximum utility an optimizer can achieve by playing against algorithm $\mathcal{A}$. We therefore say the menu $\mathcal{M}$ is *non-manipulable* if, for any optimizer payoff u_O,

$$V_O(\mathcal{M}, u_O) \leq \mathsf{Stack}(G, u_O).$$

The first point can be now rephrased as follows[9].

Theorem 4. *Fix a normal-form game G (i.e., where $\mathcal{X} = \Delta_m$ and $\mathcal{Y} = \Delta_n$). Let $\mathcal{A}$ be a no-swap-regret algorithm for G. Then the asymptotic menu $\mathcal{M}(\mathcal{A})$ is non-manipulable. Conversely, if the asymptotic menu $\mathcal{M}(\mathcal{A})$ of a no-regret algorithm $\mathcal{A}$ is non-manipulable, then the algorithm $\mathcal{A}$ must additionally be a no-swap-regret algorithm.*

We can also provide a version of Theorem 4 quantifying the degree of manipulability of a finite-horizon algorithm. To this end, we define a menu $\mathcal{M}$ to be *α-non-manipulable* if, for any payoff u_O, $V_O(\mathcal{M}, u_O) \leq \mathsf{Stack}(G, u_O) + \alpha$.

Interestingly, the relevant swap-theoretic quantity for tightly characterizing the degree of manipulability of an algorithm is not swap regret itself[10], but the *swap regret distance*, which captures the maximum distance between the menu of this algorithm and the no-swap-regret menu. Formally, for any CSP ϕ, we define $\mathsf{SwapDist}(\phi)$ to equal the maximum ℓ_2-distance $\mathrm{dist}(\phi, \mathcal{M}_{NSR})$ from ϕ to the no-swap-regret menu $\mathcal{M}_{NSR}$. Then, for any algorithm $\mathcal{A}$, we define $\mathsf{SwapDist}(\mathcal{A}) = \mathsf{SwapDist}(\mathcal{M}(\mathcal{A})) = \max_{\phi \in \mathcal{M}(\mathcal{A})} \mathsf{SwapDist}(\phi)$.

Theorem 5. *Fix a normal-form game G (i.e., where $\mathcal{X} = \Delta_m$ and $\mathcal{Y} = \Delta_n$). If $\mathcal{A}$ is a learning algorithm for G with the guarantee that* $\mathsf{SwapDist}(\mathcal{A}^T) \leq R(T)$*, then $\mathcal{M}(\mathcal{A}^T)$ is $\sqrt{mn} \cdot (R(T)/T)$-non-manipulable. Conversely, if $\mathcal{M}(\mathcal{A}^T)$ is $(R(T)/T)$-non-manipulable, then the algorithm $\mathcal{A}$ must satisfy* $\mathsf{SwapDist}(\mathcal{A}^T) \leq R(T)$.

Moving onto the second point, we say that a menu $\mathcal{M}$ is *minimal* if: i. it is the asymptotic menu of some learning algorithm $\mathcal{A}$, and ii. there is no strict sub-menu $\mathcal{M}' \subset \mathcal{M}$ which is the asymptotic menu of a learning algorithm. We can now rephrase the second point as follows.

Theorem 6. *Fix a normal-form game G (i.e., where $\mathcal{X} = \Delta_m$ and $\mathcal{Y} = \Delta_n$). All no-swap-regret algorithms $\mathcal{A}$ for G share the same asymptotic menu $\mathcal{M}(\mathcal{A}) = \mathcal{M}_{NSR}$. Moreover, if $\mathcal{A}$ is a no-regret algorithm, then $\mathcal{M}_{NSR} \subseteq \mathcal{M}(\mathcal{A})$.*

[9] The remaining theorems in this section all follow as consequences of results in [Deng et al., 2019], [Mansour et al., 2022], and [Arunachaleswaran et al., 2024]. Since they are also special cases of theorems we prove later in this paper for polytope games (Theorems 10, 11, and 12), we omit their proofs.

[10] To see why, note that the swap regret of an algorithm scales with the learner's utility function u_L, but the degree of manipulability is independent of the scale of u_L.

Finally, the result on non-manipulability (Theorem 4) concerns the utility the *optimizer* can obtain when playing against a no-swap-regret learning algorithm. But we may also wonder when it is in the interest of the learner to play such a learning algorithm. To this end, define

$$V_L(\mathcal{M}, u_O) = \max \{u_L(\phi) \mid \phi \in \mathcal{M}, u_O(\phi) = V_O(\mathcal{M}, u_O)\}$$

to be the utility the learner receives when an optimizer chooses their favorite CSP in $\mathcal{M}$ (breaking ties in favor of the learner).

Theorem 7. *Fix a normal-form game G (i.e., where $\mathcal{X} = \Delta_m$ and $\mathcal{Y} = \Delta_n$). Let $\mathcal{A}$ be a learning algorithm for G that incurs $\Omega(T)$ swap regret in the worst-case. Then there exists an optimizer utility u_O such that $V_L(\mathcal{M}_{NSR}, u_O) > V_L(\mathcal{M}(\mathcal{A}), u_O)$; i.e., there exists an optimizer against whom it is strictly better to play any no-swap-regret algorithm than $\mathcal{A}$.*

Theorem 7 can also be thought of as saying that the no-swap-regret menu is *Pareto-optimal*: there is no other learning algorithm which is asymptotically at least as good as swap regret minimization against every single possible optimizer (while strictly better for at least one optimizer). Note that unlike the other two points, this is not a tight characterization of no-swap-regret algorithms – there exist other Pareto-optimal learning algorithms that incur $\Omega(T)$ swap regret in the worst-case [Arunachaleswaran et al., 2024].

3 Profile Swap Regret

3.1 Defining Profile Swap Regret

Like polytope swap regret, profile swap regret involves performing a convex decomposition of the sequence of play of the learner into elements that we can then compute the swap regret of and aggregate. However, unlike polytope swap regret (which decomposes the strategy of the learner in each round of the game), in profile swap regret we will do this decomposition on the average CSP of play. In particular, profile swap regret can be computed as a function of just the resulting CSP ϕ of play – note that this is a property shared with linear swap regret, but that polytope swap regret and normal-form swap regret do not possess.

To define the *profile swap regret* $\mathrm{ProfSwapReg}(\phi)$ of a CSP ϕ, we perform the following steps[11]:

(1) First, decompose ϕ into a convex combination of independent (product) strategy profiles

$$\phi = \sum_{k=1}^{K} \lambda_k (x_{(k)} \otimes y_{(k)}). \tag{4}$$

for some choice of $x_{(k)} \in \mathcal{X}$, $y_{(k)} \in \mathcal{Y}$, and $\lambda_k \geq 0$ with $\sum_k \lambda_k = 1$. As with polytope swap regret, there are likely multiple ways to do this decomposition; we will eventually choose the decomposition that minimizes our eventual regret. Also note that we can freely choose the number of parts K in this decomposition.

(2) For any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, let $\mathrm{Reg}(x, y) = \max_{x^* \in \mathcal{X}} u_L(x^*, y) - u_L(x, y)$ (this can be thought of as the "instantaneous" regret from playing x in response to y). We define

[11] In Appendix D, we present an alternate formulation of profile swap regret in a way that is more directly comparable with the definition of polytope swap regret above.

$$\mathsf{ProfSwapReg}(\phi) = T \cdot \left(\min_{x_{(k)}, y_{(k)}, \lambda_k} \sum_{k=1}^{K} \lambda_k \, \mathsf{Reg}(x_{(k)}, y_{(k)}) \right), \tag{5}$$

where, as mentioned above, this minimum is over all valid convex decompositions of ϕ into the form in (4).

We will eventually demonstrate that profile swap regret is the analogue of swap regret in polytope games that preserves the game-theoretic properties mentioned in Section 2.3. Before we do so, we first remark that profile swap regret is bounded between linear swap regret and polytope swap regret.

Theorem 8. *Let* $\mathbf{x} = (x_1, x_2, \dots, x_T)$ *and* $\mathbf{y} = (y_1, y_2, \dots, y_T)$ *be a transcript of play for a given polytope game. Then* $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \le \mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) \le \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$.

Moreover, both of these inequalities are asymptotically strict in the following sense: for each inequality, there exists a family of transcripts for which the larger swap regret grows as $\Omega(T)$ *but the smaller swap regret is zero.*

Similar to the no-swap-regret menu, we will define the *no-profile-swap-regret menu* $\mathcal{M}_{NPSR}$ to be the set of CSPs ϕ satisfying $\mathsf{ProfSwapReg}(\phi) = 0$. Likewise, just as SwapDist is the relevant quantity in Theorem 5, in many of our more quantitative results it will be more convenient to work with the following "distance" variant of ProfSwapReg, where we define the *profile swap distance* $\mathsf{ProfSwapDist}(\phi) = \mathrm{dist}(\phi, \mathcal{M}_{NPSR})$ to equal the minimal Euclidean distance from ϕ to the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$.

3.2 Game-Theoretic Properties of Profile Swap Regret

We will now show that the three game-theoretic properties possessed by no-swap-regret algorithms in normal-form games – non-manipulability, minimality, and Pareto-optimality – hold for no-profile-swap-regret algorithms in general polytope games.

Our main tool to prove these results will be the same menu-based techniques used to prove Theorems 4, 6, and 7, albeit extended from the standard normal-form setting to the setting of polytope games. Underlying most of these results will be the following characterization of the no-profile-swap-regret menu as the convex hull of all "best-response CSPs" for the learner.

Lemma 9. *The no-profile-swap-regret menu* $\mathcal{M}_{NPSR}$ *is the convex hull of all CSPs of the form* $x \otimes y$ *where* $y \in \mathcal{Y}$ *and* $x \in \mathsf{BR}_L(y)$. *In particular,* $\mathsf{Stack}(G, u_O) = V_O(\mathcal{M}_{NPSR}, u_O)$.

3.2.1 Non-manipulability. We begin by establishing the analogue of Theorem 4 for polytope games: that for learning algorithms in polytope games, the property of having no-profile-swap-regret is equivalent to the property of being non-manipulable by a dynamic optimizer. At the same time, we establish the analogue of Theorem 5, providing quantitative bounds on the degree of manipulation in terms of profile swap distance (importantly, this is the quantity we will minimize in Section 4, when we turn our attention to designing no-profile-swap-regret learning algorithms).

Theorem 10. *Fix a polytope game* G. *Let* $\mathcal{A}$ *be a no-profile-swap-regret algorithm for* G. *Then the asymptotic menu* $\mathcal{M}(\mathcal{A})$ *is non-manipulable. Conversely, if the asymptotic menu* $\mathcal{M}(\mathcal{A})$ *is non-manipulable, then the algorithm* $\mathcal{A}$ *must be a no-profile-swap-regret algorithm.*

Moreover, if $\mathcal{A}$ has the guarantee that $\mathrm{ProfSwapDist}(\mathcal{A}^T) \leq R(T)$*, then* $\mathcal{M}(\mathcal{A}^T)$ *is* $\sqrt{d_L d_O}(R(T)/T)$*-non-manipulable. Conversely, if* $\mathcal{M}(\mathcal{A}^T)$ *is* $R(T)/T$*-non-manipulable, then* $\mathrm{ProfSwapDist}(\mathcal{A}^T) \leq R(T)$*.*

We briefly remark that the $\sqrt{d_L d_O}$ gap between the two bounds in Theorem 10 is an artifact of the specific boundedness constraints we placed on the optimizer's utility u_O (namely, that $||u_O||_1 \leq 1$). If we instead imposed the constraint $||u_O||_2 \leq 1$, then ProfSwapDist would be an exact measure of non-manipulability.

3.2.2 Minimality. Next, we establish the analogue of Theorem 6, proving that the no-profile-swap-regret menu is *minimal*: in particular, for any polytope game, every CSP ϕ in the no-profile-swap-regret menu is a CSP that is implementable against any no-external-regret algorithm.

Theorem 11. *Fix a polytope game G. All no-profile-swap-regret algorithms $\mathcal{A}$ for G share the same asymptotic menu $\mathcal{M}(\mathcal{A}) = \mathcal{M}_{NPSR}$. Moreover, if $\mathcal{A}$ is a no-(external)-regret algorithm, then $\mathcal{M}_{NPSR} \subseteq \mathcal{M}(\mathcal{A})$.*

3.2.3 Pareto-optimality. Finally, we establish the analogue of Theorem 7, showing that in polytope games, no-profile-swap-regret algorithms are Pareto optimal. This provides a natural counterpart to Theorem 4 (not only can no-profile-swap-regret algorithms never be manipulated by a strategizing opponent, there are always situations where the learner strictly prefers not to be manipulated).

Theorem 12. *Fix a polytope game G. Let $\mathcal{A}$ be a learning algorithm for G with* $\mathrm{ProfSwapReg}(\mathcal{A}) = \Omega(T)$*. Then there exists an optimizer utility u_O such that $V_L(\mathcal{M}_{NPSR}, u_O) > V_L(\mathcal{M}(\mathcal{A}), u_O)$; i.e., there exists an optimizer against whom it is strictly better to play any no-profile-swap-regret algorithm than $\mathcal{A}$.*

4 Minimizing Profile Swap Regret

We now switch our attention to the problem of designing learning algorithms that minimize profile swap regret over finite time horizons. More specifically, we are interested in the following two questions:

(1) What is the best bound on profile swap regret that we can guarantee after T rounds? Can we guarantee that this quantity is sublinear in T and polynomial in the dimensions d_L and d_O?

(2) Can we construct *computationally efficient* algorithms with these regret bounds? That is, can we construct learning algorithms that run in per-iteration time that is polynomial in T, d_L, and d_O?

Note that for linear swap regret, we have positive answers to both of these questions. In particular, linear swap regret is known to be tractable to minimize: there is a line of work [Dann et al., 2023, Farina and Pipis, 2023, 2024, Gordon et al., 2008] that provides efficient learning algorithms achieving $\mathrm{poly}(d)\sqrt{T}$ linear swap regret in specific classes of games, with [Daskalakis et al., 2024a] now showing that this is possible as long as you have standard oracle access to the learner's action set $\mathcal{X}$. On the other side of the spectrum, the best known algorithms for minimizing normal-form swap regret generally incur regret bounds that are either polynomial in the number of *vertices* of $\mathcal{X}$ (which easily can be exponential in dimension for Bayesian games and extensive-form games) or scale as $T/(\log T)^{O(1)}$ (i.e., requiring exponential in $1/\varepsilon$ rounds to guarantee εT regret). Recent lower bounds [Daskalakis et al., 2024b] show that these rates are necessary for minimizing normal-form swap regret in extensive-form games, implying a mostly negative answer to the above questions.

4.1 Information-theoretic regret bounds via Blackwell approachability

In contrast to normal-form swap regret, we will show that it is possible to design learning algorithms that incur at most $O(\sqrt{T})$ profile swap regret. Our main technique is to frame the problem of minimizing profile swap regret as a specific instance of Blackwell approachability where the goal is to approach the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$.

We therefore begin by briefly reviewing the standard theory of Blackwell approachability. An instance of Blackwell approachability is specified by a convex action set $\mathcal{X}$ for the learner, a convex action set $\mathcal{Y}$ for the adversary, a bilinear *vector-valued* payoff function $v : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, and a convex target set $\mathcal{S} \subseteq \mathbb{R}^d$. The goal of the learner is to run a learning algorithm $\mathcal{A}$ (outputting a sequence of values x_t in response to the actions $y_1, y_2, \dots, y_{t-1}$ played by the adversary so far) that guarantees that the average payoff $\frac{1}{T}\sum_{t=1}^{T} v(x_t, y_t)$ is close to the target set $\mathcal{S}$.

In order for this to be possible at all, the function v must be *response-satisfiable* with respect to the target set $\mathcal{S}$: that is, for every possible action $y \in \mathcal{Y}$ played by the adversary, there must be a response $x(y) \in \mathcal{X}$ for the learner with the property that $v(x(y), y) \in \mathcal{S}$. If v is not response-satisfiable for some y, then if the adversary repeatedly plays this action, it is impossible for the learner to force the average payoff to converge to the set $\mathcal{S}$. Blackwell's approachability theorem states that response-satisfiability is the only necessary requirement – if v is response-satisfiable with respect to $\mathcal{S}$, then it is always possible for the learner to steer the average payoff to approach $\mathcal{S}$. Below we cite a quantitative, algorithmic version of Blackwell's approachability theorem that appears in [Mannor and Perchet, 2013].

Theorem 13 (Blackwell's approachability theorem; Proposition 1 of [Mannor and Perchet, 2013]). *Let $v\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be a vector-valued bilinear payoff function and $\mathcal{S}$ be a convex subset of $\mathbb{R}^d$ such that v is response-satisfiable with respect to $\mathcal{S}$. Then there exists a learning algorithm $\mathcal{A}$ (choosing $x_t \in \mathcal{X}$ in response to $y_1, y_2, \dots, y_{t-1} \in \mathcal{Y}$) with the property that:*

$$T \cdot \operatorname{dist}\left(\frac{1}{T}\sum_{i=1}^{t} v(x_t, y_t), \mathcal{S}\right) \leq 2\|v\|_\infty \sqrt{T},$$

where $\|v\|_\infty = \max_{x\in\mathcal{X}, y\in\mathcal{Y}} \|v(x,y)\|_2$ and $\operatorname{dist}(z, \mathcal{S})$ is the minimum (ℓ_2) distance from x to the point $\mathcal{S}$.

Moreover, this algorithm can be implemented efficiently (in time polynomial in d, $\dim(\mathcal{X})$, and $\dim(\mathcal{Y})$) given an efficient separation oracle for the set $\mathcal{S}$.

There is a clear similarity between the Blackwell approachability problem stated above and the problem of minimizing profile swap regret: whereas in Blackwell approachability, the learner wants to guide the average vector-valued payoff $\frac{1}{T}\sum v(x_t, y_t)$ to converge to the set $\mathcal{S}$, when minimizing profile swap regret, the learner wants to guide the average CSP $\frac{1}{T}\sum x_t \otimes y_t$ to converge to the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$. In fact, the function mapping a pair (x, y) of learner and optimizer strategies to the corresponding CSP $x \otimes y$ is itself a vector-valued bilinear payoff function that is response-satisfiable with respect to the no-profile swap regret menu. This allows us to immediately apply Theorem 13 to get bounds on profile swap regret, which we do below (note that this Theorem will mostly be later subsumed by Theorem 20, which achieves similar bounds but efficiently).

Theorem 14. *For any polytope game G (where $\mathcal{X}$ and $\mathcal{Y}$ are bounded in unit norm), there exists a learning algorithm $\mathcal{A}$ that guarantees* $\mathsf{ProfSwapDist}(\mathcal{A}^T) \leq 2\sqrt{T}$.

4.2 Hardness of computing profile swap regret

Theorem 14 indicates that it is possible to construct a learning algorithm that achieves good profile swap regret guarantees, but states nothing about the computational complexity of running such an algorithm. The instantiation of Blackwell's approachability in Theorem 13 is algorithmic, and can be run in polynomial-time as long as one has efficient oracle access to the target set in question.

In our case, this target set is the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$. In some cases, we can construct efficient convex oracles for the set $\mathcal{M}_{NPSR}$; for example, if our polytope game is actually a normal-form game, the menu $\mathcal{M}_{NPSR}$ reduces to the no-swap-regret polytope $\mathcal{M}_{NSR}$, which can be written as the explicit intersection of $O(n^2)$ half-spaces when the learner has n actions (i.e., constraints guaranteeing the learner cannot improve their utility by swapping action $i \in [n]$ to action $i' \in [n]$).

However, we will show that for general polytope games, we cannot hope for any computationally efficient characterization of the no-profile-swap-regret menu (even when we have succinct and efficient descriptions of the sets $\mathcal{X}$ and $\mathcal{Y}$). In particular, we will show that even for the setting of Bayesian games, it is impossible to even compute the profile swap distance $\mathsf{ProfSwapDist}(\phi)$ (note that this quantity is simply the distance from ϕ to $\mathcal{M}_{NPSR}$, and would be efficient to compute given standard convex oracles for this set). Our main tool is the following APX-hardness result of computing Stackelberg equilibria in Bayesian games.

Lemma 15 (Theorem 14 in [Mansour et al., 2022])**.** *It is APX-hard to compute the Stackelberg value for the optimizer in a Bayesian game. That is, there exists a constant $\varepsilon > 0$ such that given a Bayesian game G (with a specific optimizer utility u_O) and a value $V > 0$ it is NP-hard to distinguish between the cases* $\mathsf{Stack}(G, u_O) \leq (1-\varepsilon)V$ *and* $\mathsf{Stack}(G, u_O) \geq V$.

Lemma 15 almost immediately rules out the possibility of weak optimization oracles for $\mathcal{M}_{NPSR}$, since by Lemma 9, the Stackelberg value $\mathsf{Stack}(G, u_O)$ is simply the maximum value u_O takes over the menu $\mathcal{M}_{NPSR}$. Since most weak convex oracles are equivalent (given mild assumptions), this allows us to rule out the existence of any algorithm that can efficiently compute $\mathsf{ProfSwapDist}(\phi)$.

Theorem 16. *Define a* profile swap distance oracle *to be an algorithm that takes as input a Bayesian game G (with $\mathcal{X} = \Delta_m^{c_L}$ and $\mathcal{Y} = \Delta_n^{c_O}$), a CSP $\phi \in \mathcal{X} \otimes \mathcal{Y}$, and a precision parameter δ, and returns a δ-additive approximation to* $\mathsf{ProfSwapDist}(\phi)$. *If $P \neq NP$, then there does not exist a profile swap distance oracle that runs in time* $\mathrm{poly}(n, m, c_L, c_O, 1/\delta)$.

4.3 Efficient learning algorithms via semi-separation

Nevertheless, despite the hardness of computing profile swap regret (and thus of running the algorithm of Theorem 14), we will now show that it is still possible to design efficient learning algorithms $\mathcal{A}$ that obtain profile swap regret guarantees of the form $\mathrm{poly}(d_L, d_O)\sqrt{T}$.

To describe the idea behind semi-separation, it will be convenient to express the problem of Blackwell approachability in slightly different language. Previously, we described an approachability instance as being parameterized by (in addition to the two action sets of the learner and adversary) a vector-valued bilinear payoff function v and a target set $\mathcal{S}$, with the idea that the learner is trying to steer the average vector-valued payoff to be close to $\mathcal{S}$. In learning-theoretic contexts, it is frequently mathematically more convenient to work with another formulation of approachability that we will refer to as *orthant-approachability*. An *orthant-approachability* instance is parameterized by a single *convex set* $\mathcal{U}$ of *one-dimensional* biaffine payoff functions $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The goal of the learner is now to minimize the worst-case average value of $u(x_t, y_t)$ over *all* biaffine functions $u \in \mathcal{U}$; that is, they wish to minimize the approachability loss given by $\mathrm{AppLoss}(\mathbf{x}, \mathbf{y}) = \max_{u \in \mathcal{U}} \frac{1}{T} \sum_{t=1}^{T} u(x_t, y_t)$.

In particular, the learner would like to guarantee that $\mathrm{AppLoss}(\mathbf{x}, \mathbf{y})$ grows sublinearly in T. In order for this to be possible, $\mathcal{U}$ must contain only *response-satisfiable* functions u. In the language of orthant-approachability, this means that for any $y \in \mathcal{Y}$, there exists an $x \in \mathcal{X}$ such that $u(x, y) \leq 0$ for all $u \in \mathcal{U}$ (we call this "orthant"-approachability since this can be thought of approaching the negative orthant for the infinite-dimensional bilinear payoff function whose entries are given by $u(x, y)$ for different $u \in \mathcal{U}$).

It can be shown that orthant-approachability and our previous formulation of approachability are essentially equivalent (in fact, orthant-approachability is slightly more general, in that it can capture other distance metrics and some infinite-dimensional bilinear payoffs). In particular, we show below (following an argument in [Dann et al., 2024]) that any instance of our previous formulation of approachability can be equivalently expressed as an orthant-approachability problem.

Lemma 17. *Let $v : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be a vector-valued bilinear payoff function that is response-satisfiable with respect to the target set $\mathcal{S} \subseteq \mathbb{R}^d$. There exists a convex subset $\mathcal{U}$ of response-satisfiable biaffine functions $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that, for any sequences $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$,*

$$\max_{u \in \mathcal{U}} \frac{1}{T} \sum_{t=1}^{T} u(x_t, y_t) = \mathrm{dist}\left(\frac{1}{T} \sum_{i=1}^{t} v(x_t, y_t), \mathcal{S} \right)$$

As with classical Blackwell approachability, there exist efficient learning algorithms for minimizing approachability loss in orthant-approachability – however, just as the earlier algorithms required separation oracles for the target set $\mathcal{S}$, the algorithms for orthant-approachability require separation oracles for the set of payoff functions $\mathcal{U}$. And likewise, just as Theorem 16 precludes the existence of efficient separation oracles for the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$, it also precludes the existence of separation oracles for the corresponding convex set $\mathcal{U}$.

[Daskalakis et al., 2024a] bypass a similar obstacle for minimizing linear swap regret via a technique they call *semi-separation*. We introduce a version of this same technique here for orthant-approachability. To motivate this technique, note that if we want to achieve orthant-approachability with respect to a specific (perhaps intractible) set of biaffine functions $\mathcal{U}$, it suffices to achieve orthant-approachability with respect to any superset $\mathcal{U}'$ containing $\mathcal{U}$ (since the maximum over all u' in $\mathcal{U}'$ will always be at least as large as the maximum over all u in $\mathcal{U}$). In particular, if we could find a superset $\mathcal{U}'$ of $\mathcal{U}$ with more tractable convex oracles, then we could simply run orthant-approachability over that superset.

Unfortunately, it is not too hard to show that the minimality of the menu $\mathcal{M}_{NPSR}$ implies that no superset $\mathcal{U}'$ of $\mathcal{U}$ can contain only response-satisfiable functions. The idea of [Daskalakis et al., 2024a] is to run an approachability algorithm on some tractable superset $\mathcal{U}'$ anyway, despite $\mathcal{U}'$ not being an approachable set. The approachability algorithm will either work without issue (in which case it will provide our desired guarantee on the approachability loss), or at some point the algorithm will identify a $u' \in \mathcal{U}'$ that is not response-satisfiable, and thus not in $\mathcal{U}$. If we can then construct a separating hyperplane that separates u' from $\mathcal{U}$, we can use this to refine our superset $\mathcal{U}'$. This algorithm – which takes a non-response-satisfiable u, and returns a hyperplane separating u from $\mathcal{U}$ – is what we refer to as a semi-separation oracle (unlike a standard separation oracle, it does not work for all $u \notin \mathcal{U}$, but only non-response-satisfiable u).

Formally, a *semi-separation oracle* for a set of response-satisfiable payoff functions $\mathcal{U}$ is an algorithm which takes an arbitrary bi-affine function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and either (i) returns that the function u is response-satisfiable, or (ii) returns a hyperplane separating u from $\mathcal{U}$. The following theorem states that efficient semi-separation oracles suffice for performing orthant-approachability efficiently. The proof closely follows the proof of [Daskalakis et al., 2024a] for the case of linear swap functions, and is deferred to Appendix E.

Theorem 18 (Approachability with semi-separation oracles). *Consider an orthant-approachability instance $(\mathcal{X}, \mathcal{Y}, \mathcal{U})$ where $\mathcal{X} \subseteq \mathcal{B}(D_{\mathcal{X}})$, $\mathcal{Y} \subseteq \mathcal{B}(D_{\mathcal{Y}})$, and $\mathcal{B}(u_0, \rho) \subseteq \mathcal{U} \subseteq \mathcal{B}(D)$ (for some known u_0 and radius ρ). If we have access to efficient separation oracles for $\mathcal{X}$ and $\mathcal{Y}$, and an efficient semi-separation oracle for the set $\mathcal{U}$, then Algorithm 1 has the guarantee that*

$$\max_{u \in \mathcal{U}} \frac{1}{T} \sum_{t=1}^{T} u\left(x_t, y_t\right) \leq O\left(\frac{D_{\mathcal{X}} D_{\mathcal{Y}} D}{\sqrt{T}}\right),$$

and runs in time $\mathrm{poly}(D_{\mathcal{X}}, D_{\mathcal{Y}}, D, \rho^{-1}, \dim(\mathcal{U}), T)$.

In the remainder of this section, we will show that we can construct a semi-separation oracle for the approachability problem corresponding to minimizing profile swap regret, thus allowing us to efficiently minimize it via Theorem 18. In particular, consider the set $\mathcal{U}$ containing all bi-affine functions $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ of the form $u(x, y) = \langle h, x \otimes y \rangle - b$ that satisfy: i. $\|h\| \leq 1$, ii. $|b| \leq 1$, and iii. $u(x, y) \leq 0$ for all $x \in \mathrm{BR}_L(y)$. Note that this is a convex set of bi-affine functions, is response-satisfiable (via the third constraint), and for any CSP ϕ satisfies $\max_{u \in \mathcal{U}} u(\phi) = \mathsf{ProfSwapDist}(\phi)$ (in particular, it is equivalent to the set constructed in the proof of Lemma 17). We produce an efficient semi-separation oracle for this choice of $\mathcal{U}$.

Lemma 19. *Given efficient separation oracles for the sets $\mathcal{X}$ and $\mathcal{Y}$, it is possible to construct an efficient semi-separation oracle for the set $\mathcal{U}$ defined above.*

From this semi-separation oracle and Theorem 18, we immediately obtain an efficient algorithm for minimizing profile swap regret.

Theorem 20. *Given efficient separation oracles for $\mathcal{X}$ and $\mathcal{Y}$, there exists a learning algorithm $\mathcal{A}$ such that* $\mathsf{ProfSwapDist}(\mathcal{A}^T) = O(\sqrt{T})$ *that runs in* $\mathrm{poly}(T, d_L, d_O)$ *time per iteration.*

References

Jacob Abernethy, Peter L Bartlett, and Elad Hazan. Blackwell approachability and no-regret learning are equivalent. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 27–46. JMLR Workshop and Conference Proceedings, 2011.

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. Pareto-optimal algorithms for learning in games. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 490–510, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400707049. doi: 10.1145/3670865.3673517. URL https://doi.org/10.1145/3670865.3673517.

Robert J Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 523–538, 2018.

Noam Brown and Tuomas Sandholm. Libratus: The superhuman AI for no-limit poker. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 5226–5228. ijcai.org, 2017.

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. *arXiv preprint arXiv:2310.19786*, 2023.

Christoph Dann, Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Pseudonorm approachability and applications to regret minimization. In *International Conference on Algorithmic Learning Theory*, pages 471–509. PMLR, 2023.

Christoph Dann, Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Rate-preserving reductions for blackwell approachability. *arXiv preprint arXiv:2406.07585*, 2024.

Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Charilaos Pipis, and Jon Schneider. Efficient learning and computation of linear correlated equilibrium in general convex games. *arXiv preprint arXiv:2412.20291*, 2024a.

Constantinos Daskalakis, Gabriele Farina, Noah Golowich, Tuomas Sandholm, and Brian Hu Zhang. A lower bound on swap regret in extensive-form games, 2024b. URL https://arxiv.org/abs/2406.13116.

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. *Advances in Neural Information Processing Systems*, 32, 2019.

Gabriele Farina and Charilaos Pipis. Polynomial-time Linear-Swap Regret Minimization in Imperfect-Information Sequential Games. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Gabriele Farina and Charilaos Pipis. Polynomial-Time Computation of Exact Φ-Equilibria in Polyhedral Games, 2024. URL https://arxiv.org/abs/2402.16316.

Gabriele Farina, Ioannis Anagnostides, Haipeng Luo, Chung-Wei Lee, Christian Kroer, and Tuomas Sandholm. Near-optimal no-regret learning dynamics for general convex games. In *Neural Information Processing Systems (NeurIPS)*, 2022.

Dean P Foster and Rakesh V Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2): 40, 1997.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In Avrim Blum and Michael J. Kearns, editors, *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996*, pages 325–332. ACM, 1996.

Kaito Fujii. Bayes correlated equilibria and no-regret dynamics. *arXiv preprint arXiv:2304.05005*, 2023.

Geoffrey J. Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *Proceedings of ICML*, volume 307, pages 360–367. ACM, 2008.

Yin Tat Lee, Aaron Sidford, and Santosh S Vempala. Efficient convex optimization with membership oracles. In *Conference On Learning Theory*, pages 1292–1294. PMLR, 2018.

Shie Mannor and Vianney Perchet. Approachability, fast and slow. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 30, pages 1–16, 2013.

Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in bayesian games. In *Conference on Learning Theory*, pages 5221–5252. PMLR, 2022.

Matej Moravcík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael H. Bowling. Deepstack: Expert-level artificial intelligence in no-limit poker. *CoRR*, abs/1701.01724, 2017.

Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):1–29, 2008.

Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1223–1234, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703836. doi: 10.1145/3618260.3649691. URL https://doi.org/10.1145/3618260.3649691.

Aviad Rubinstein and Junyao Zhao. Strategizing against no-regret learners in first-price auctions. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 894–921, 2024.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016.

Bernhard Von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

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*, 2024.

A Game-Agnostic Learning and Polytope Swap Regret

In our discussion of online learning thus far, we have assumed that our learning algorithms $\mathcal{A}$ are tailored to a specific polytope game G. By this, we mean that our algorithms operate under the assumption that the learner can see the sequence of actions $y_t \in \mathcal{Y}$ played by the optimizer so far and that the learner can compute the game-specific payoff function $u_L(x, y)$. These assumptions are typical for applications of online learning to games (for example, when such algorithms are used for equilibrium computation in extensive-form games).

However, many algorithms in the field of adversarial online learning are designed so that they require only the counterfactual rewards (alternatively, losses) faced by the learner over time. That is, instead of being told the mixed strategy y_t the optimizer played in round t, it is enough for these learning algorithms to have access to the linear function $r_t(x)$ describing what utility the learner would have received if they had played action $x \in \mathcal{X}$ that round. Such algorithms can be used by a learner to play any repeated game where the learner has action set $\mathcal{X}$, regardless of their specific payoff function u_L or the optimizer's action space $\mathcal{Y}$. In this section, we explore what happens when we restrict ourselves to game-agnostic learning algorithms – we will show that profile swap regret is an inherently game-dependent notion, and if we wish to minimize profile swap regret with a game-agnostic learning algorithm, we must actually minimize polytope swap regret.

Formally, a *game-agnostic learning algorithm* $\mathcal{A}$ is defined in the same way as our earlier (game-dependent) learning algorithms, with the only difference being that in each horizon-dependent learning algorithm, $\mathcal{A}_t^T$ now maps a sequence of affine linear *reward functions* $r_1, r_2, \dots, r_{t-1}$ to the action $x_t \in \mathcal{X}$ the learner will take at time t. Each r_t is an affine linear function sending $\mathcal{X}$ to $[-1, 1]$; we will write $\mathcal{X}^*$ to denote the set of such functions (so, each r_t belongs to $\mathcal{X}^*$).

When used to play a specific polytope game G, the function r_t is constructed via $r_t(x) = u_L(x, y_t)$, where $y_t \in \mathcal{Y}$ is the action of the optimizer at round t. In particular, any transcript $(\mathbf{x}, \mathbf{y})$ of a polytope game G corresponds to a *game-agnostic transcript* $(\mathbf{x}, \mathbf{r})$ via this mapping. The game-agnostic transcript is sufficient to compute many quantities relevant to the learner, including their total utility over the course of the game and several variants of regret, as shown in the following lemma.

Lemma 21. *Let* $(\mathbf{x}, \mathbf{y})$ *be the transcript of some repeated polytope game* G*. It is possible to compute external regret* $\mathsf{Reg}(\mathbf{x}, \mathbf{y})$*, linear swap regret* $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y})$*, and polytope swap regret* $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$ *from the game-agnostic transcript* $(\mathbf{x}, \mathbf{r})$ *corresponding to* $(\mathbf{x}, \mathbf{y})$ *(without any knowledge of* G *or* u_L *aside from the learner's action set* $\mathcal{X}$*).*

As a consequence of Lemma 21, we can define quantities such as $\mathsf{Reg}(\mathbf{x}, \mathbf{r})$, $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{r})$, and $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{r})$ (defining these swap regrets as functions of the game-agnostic transcript). Similarly, we can define e.g. $\mathsf{PolySwapReg}(\mathcal{A}^T)$ to be the maximum polytope swap regret incurred by the game-agnostic (horizon T) algorithm $\mathcal{A}^T$ against the worst-case sequence of reward functions $\mathbf{r}$.

In contrast to the result of Lemma 21, it is *not possible* to compute the profile swap regret $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y})$ in this way: there exist pairs of transcripts $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2)$ for polytope games with the same action set $\mathcal{X}$ that have the same game-agnostic transcript $(\mathbf{x}, \mathbf{r})$, but where $\mathsf{ProfSwapReg}(\mathbf{x}_1, \mathbf{y}_1) \neq \mathsf{ProfSwapReg}(\mathbf{x}_2, \mathbf{y}_2)$. That is, unlike the other variants of swap regret we have discussed, profile swap regret is fundamentally game-dependent (intuitively, this is because the space of possible decompositions of the CSP ϕ depends on the optimizer's action space $\mathcal{Y}$). We give an explicit example of this phenomenon in Appendix F.1.

Instead, for any specific polytope game G and game-agnostic learning algorithm $\mathcal{A}$ (where $\mathcal{A}$ and G share the same learner action set $\mathcal{X}$), we can define $\mathsf{ProfSwapReg}_G(\mathcal{A}^T)$ to be the maximum profile swap regret a learner incurs by using $\mathcal{A}^T$ to select their actions in the repeated game G. We will show that if we want to bound the profile swap regret $\mathsf{ProfSwapReg}_G(\mathcal{A}^T)$ for all polytope games G, we must bound the (game-agnostic) polytope swap regret $\mathsf{PolySwapReg}(\mathcal{A}^T)$. Our main tool is the following lemma, showing that we can always instantiate any game-agnostic transcript as a transcript of an actual polytope game in a way that makes profile swap regret and polytope swap regret agree.

Lemma 22. *Let* $(\mathbf{x}, \mathbf{r})$ *be a game-agnostic transcript. There exists a transcript* $(\mathbf{x}, \mathbf{y})$ *of a polytope game* G *where* $(\mathbf{x}, \mathbf{r})$ *is the game-agnostic transcript corresponding to* $(\mathbf{x}, \mathbf{y})$ *and* $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$.

With Lemma 22, we can prove the main result of this section.

Theorem 23. *Let* $\mathcal{A}$ *be a game-agnostic learning algorithm. Then for any* $T > 0$, $\mathsf{PolySwapReg}(\mathcal{A}^T) = \max_G \mathsf{ProfSwapReg}_G(\mathcal{A}^T)$, *where this maximum is over all polytope games* G *with the same learner action set* $\mathcal{X}$ *as* $\mathcal{A}$.

Remark 1. It is interesting to discuss the results of [Rubinstein and Zhao, 2024] in light of the above discussion. [Rubinstein and Zhao, 2024] answer an open question of [Mansour et al., 2022] by showing that minimizing *polytope swap regret* is both necessary and sufficient for implying non-manipulability in Bayesian games. However, the original question of [Mansour et al., 2022] was posed for (a slight variant of) game-agnostic learning algorithms – indeed, [Rubinstein and Zhao, 2024] point out that without the ability to change the Bayesian game (by increasing the number of actions for the optimizer), the conjecture is not true (we use a similar construction to separate profile swap regret and polytope swap regret in Theorem 8). Our Theorem 23 above (along with the characterization of non-manipulability in Theorem 10) can therefore be thought of as a generalization of this result of [Rubinstein and Zhao, 2024] to all polytope games.

B Implications for Equilibrium Computation

We finally return to the question of equilibrium computation. Indeed, the original motivation for introducing swap regret was to design learning dynamics that converge to the notion of correlated equilibria in normal-form games (e.g., see [Foster and Vohra, 1997]). It seems natural then, that when defining swap regret for polytope games, we should choose a quantity whose minimization guarantees convergence to the set of correlated equilibria.

The problem here is that, just as it is not exactly clear what the definition of swap regret should be in polytope games, it is also not exactly clear what the definition of correlated equilibrium should be in polytope games. Even for the restricted set of Bayesian games, existing correlated equilibrium notions include linear correlated equilibria, normal-form correlated equilibria, agent-normal-form correlated equilibria, and communication equilibria [Fujii, 2023]. This raises the question of why we might want to compute correlated equilibria in the first place – what properties might we desire from its definition?

One motivation for studying correlated equilibria in normal-form games, arising from the original definition in [Aumann, 1974], is that we can view a correlated equilibrium as an outcome that can be implemented by a third-party *mediator* who privately recommends an action to each player. If no one can gain by deviating from these suggestions, the resulting strategy profile constitutes a correlated equilibrium. This gives a definition of correlated equilibrium that is particularly amenable to mechanism design (the classic example here is that of a traffic light coordinating the actions of many cars at an intersection).

This mediator-based definition is relatively straightforward to extend to the setting of a polytope game G – we are looking for distributions over joint recommendations $(x, y) \in \mathcal{X} \otimes \mathcal{Y}$ with the property that neither player has an incentive to deviate after seeing their recommendation. It can be shown (see Appendix G.1) that these recommendations can, without loss of generality, be supported on the vertices of $\mathcal{X}$ and $\mathcal{Y}$, and so these mediator-based correlated equilibria are exactly the *normal-form correlated equilibria (NFCE)* of G (equivalently, these are the correlated equilibria of the normal-form vertex game G^V). Formally, an NFCE of G is a vertex game CSP $\phi^V \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ with the property that[12] $\mathsf{NFSwapReg}_X(\phi^V) = \mathsf{NFSwapReg}_Y(\phi^V) = 0$. Since this game can be thought of as a normal-form game over the action set simplices $\Delta(\mathcal{V}(\mathcal{X}))$ and $\Delta(\mathcal{V}(\mathcal{Y}))$, $\mathsf{NFSwapReg}_X(\phi^V)$ is simply defined as the swap-regret of a CSP ϕ^V in this game.

But a second, learning-theoretic, motivation for studying correlated equilibria is that they directly represent possible summaries of the outcomes of learning dynamics in repeated games. These are useful for understanding what possible outcomes we might expect from repeated multi-agent interactions in strategic settings – after all, it is now generally accepted that it is intractable (computationally and otherwise) for players to play according to the Nash equilibrium of an arbitrary general-sum game, and instead it is more reasonable to model players as performing some form of learning to decide their actions over time.

From this perspective, the definition of correlated equilibrium should follow from the definition of swap regret (specifically, whichever definition of swap regret we choose to model rational behavior in repeated games). For profile swap regret, this gives rise to the notion of *profile correlated equilibria (PCE)*. In particular, a CSP ϕ is a PCE of a polytope game G if $\mathsf{ProfSwapReg}_X(\phi) = \mathsf{ProfSwapReg}_Y(\phi) = 0$.

In normal-form games, these two motivations give rise to exactly the same notion of correlated equilibrium (and hence this distinction often goes unmade). In polytope games these two notions are not immediately comparable – a normal-form CE ϕ^V and a profile CE ϕ belong to different sets and have different dimensions. However, every normal-form CE $\phi^V \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ naturally corresponds to a CSP $\mathrm{Proj}(\phi^V)$ obtained by sending ϕ^V to the element $\sum_{v \in \mathcal{V}(\mathcal{X}), w \in \mathcal{V}(\mathcal{Y})} \phi^V(v, w)(v \otimes w)$. This raises the question: are profile correlated equilibria exactly the CSPs corresponding to valid normal-form correlated equilibria?

To see why we might expect this to be the case for profile swap regret in particular, we point out that a one-sided version of this question has a positive answer. In particular, every CSP ϕ that incurs zero profile swap regret for the $\mathcal{X}$-player (i.e., satisfies $\mathsf{ProfSwapReg}_X(\phi) = 0$) can be instantiated as $\mathrm{Proj}(\phi^V)$ for some vertex-game CSP ϕ^V that incurs zero normal-form swap regret for the $\mathcal{X}$-player (see Appendix G.3). From a mediator point-of-view, this means that any CSP ϕ the $\mathcal{X}$-player encounters by playing a no-profile-swap-regret learning algorithm can also be induced by a "one-sided mediator" that only needs to consider the incentives of the $\mathcal{X}$-player (i.e., one that the $\mathcal{Y}$-player will blindly trust). We note that this property follows from the fact that the no-profile-swap-regret menu is minimal (Theorem 11), and does not hold for weaker forms of regret like linear swap regret.

[12] Since in this section we are concerned with settings where both players are learning, instead of referring to the two players as the "learner" and "optimizer", we will refer to them as the "$\mathcal{X}$-player" and "$\mathcal{Y}$-player", adjusting subscripts accordingly.

However, we prove that the answer to the original question is no. Specifically, we show that although every CSP $\mathrm{Proj}(\phi^V)$ corresponding to a normal-form CE ϕ^V is in fact a profile CE, there exist profile CE that cannot be written in this form. In fact, we go further and show that there are profile CE with utility profiles that no normal-form CE can generate.

Theorem 24. *If ϕ^V is a normal-form CE in a polytope game G, then $\mathrm{Proj}(\phi^V)$ is a profile CE in G. On the other hand, there exists a polytope game G and a profile CE ϕ such that there does not exist a normal-form CE ϕ^V satisfying $\mathrm{Proj}(\phi^V) = \phi$ (in fact, there does not even exist a normal-form CE ϕ^V where $u_X(\phi^V) = u_X(\phi)$ and $u_Y(\phi^V) = u_Y(\phi)$).*

In particular, Theorem 24 means that there are outcomes of repeated polytope games (i.e., CSPs ϕ) for which both players can imagine as implementable by a one-sided mediator, but for which no (two-sided) mediator protocol can actually induce. This is a fundamental property of polytope games that extends beyond the specific definition of profile swap regret, and distinguishes general polytope games from normal-form games.

Of course, we can also ask for which polytope games G does a separation exist like that in Theorem 24 – after all, there is no separation for the case of normal-form games, which are special cases of polytope games. Fully characterizing this is an interesting open question, but below we show that this gap disappears whenever *either* player's action set is a simplex. Notably, this covers all standard Bayesian games where only one of the two players has any private information (e.g., some auction/pricing games).

Theorem 25. *Let G be a polytope game where $\mathcal{X}$ is a simplex (but where $\mathcal{Y}$ may be any polytope). Then, if ϕ is a profile CE in G, there exists a normal-form CE ϕ^V for which $\mathrm{Proj}(\phi^V) = \phi$.*

Finally, we conclude with a couple of remarks on the problem of actually computing equilibria. Note that our efficient no-profile-swap-regret learning algorithm (Theorem 20) immediately gives us an efficient algorithm to compute *some* profile CE in any (efficiently representable) polytope game G, by making both players run these low-profile-swap-regret dynamics (see Appendix G.4). On the other hand, if we want to *optimize* over the set of profile CE, we need to contend with the hardness result in Theorem 16 – in fact, by setting the utilities of one of the players to zero (so that they are indifferent between all outcomes), the set of profile CE becomes exactly the no-profile-swap-regret menu, which is provably hard to optimize over. This situation is reminiscent of the situation in [Papadimitriou and Roughgarden, 2008] for computing correlated equilibria in succinct multiplayer games (although note that here this phenomenon occurs for games with only two players).

C Additional Related Work

Swap regret has long been an object of interest in normal-form games. Early methods to achieve bounded swap regret were established by [Blum and Mansour, 2007], who proposed efficient algorithms that guarantee low internal regret. More recently, concurrent work by [Dagan et al., 2023] and [Peng and Rubinstein, 2024] have shown that it is possible to minimize swap regret whenever it is possible to minimize external regret in normal-form games.

While no-swap regret algorithms were first conceptualized via their connection to correlated equilibrium, a more recent line of work has investigated the strategic properties of these algorithms in their own right. [Braverman et al., 2018] initiated the study of non-myopic responses to learning algorithms in the context of games. In this setting, [Deng et al., 2019] showed that no-swap regret algorithms are *non-manipulable*, and thus optimizing against a no-swap regret algorithm involves optimizing only over the class of static strategies that do not vary over time.

[Mansour et al., 2022] introduced the study of non-myopic responses to learning algorithms in *Bayesian* games and Polytope games more generally, identifying Polytope Swap Regret as a sufficient notion for non-manipulability in these games. [Rubinstein and Zhao, 2024] prove that Polytope Swap Regret is also *necessary* in general Bayesian games, but in the game-agnostic setting. By contrast, we show that when the structure of the game is fixed, the strictly weaker notion of *profile* swap regret is both necessary and sufficient for all Polytope games.

The concept of correlated equilibria originates in [Aumann, 1974]; who introduced this notion to generalize the notion of Nash Equilibria by adding an external signal generating mediator who helps the players correlate their play via signals. Correlated Equilibria have a deep connection to no-swap-regret algorithms. [Foster and Vohra, 1997] showed that players using calibrated learning algorithms in games converge to a correlated equilibrium. This line of research (including [Blum and Mansour, 2007]) has noted that swap-regret minimization (a consequence of calibrated learning) suffices to guarantee convergence to correlated equilibria. Specifically, in normal form games, every correlated equilibria (defined with an external mediator) can be reached by the interplay of some no-swap-regret algorithms.

D Alternate formulation of profile swap regret

In this appendix, we present an alternate formulation of profile swap regret with some advantages for parsing this as a "swap regret" notion and comparing it to other notions of swap regret (i.e., the definition below is used in the that profile swap regret is upper bounded by polytope swap regret in Theorem 8).

We begin by showing that in the original definition of profile swap regret, it suffices to consider decompositions (of the form (4)) where each $x_{(k)}$ is a vertex of $\mathcal{X}$.

Lemma 26. *Let $\phi = \sum_{k=1}^{K} \lambda_k (x_{(k)} \otimes y_{(k)})$ be an arbitrary decomposition of the CSP ϕ. Then there exists another decomposition $\phi = \sum_{k=1}^{K'} \lambda'_k (x'_{(k)} \otimes y'_{(k)})$ where each $x'_{(k)} \in \mathcal{V}(\mathcal{X})$, all $x'_{(k)}$ are distinct, and*

$$\sum_{k=1}^{K'} \lambda'_k \operatorname{Reg}(x'_{(k)}, y'_{(k)}) \leq \sum_{k=1}^{K} \lambda_k \operatorname{Reg}(x_{(k)}, y_{(k)}). \tag{6}$$

One consequence of Lemma 26 is that it allows us to write profile swap regret as a "swap regret": that is, regret with respect to the set of swap functions $\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})$ (swapping vertices of $\mathcal{X}$), something not immediately clear from the original definition. In particular, we can equivalently define the profile swap regret ProfSwapReg(ϕ) of a CSP via the following steps:

(1) Decompose ϕ into a convex combination of strategy profiles of the form

$$\phi = \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v (v \otimes y_v), \tag{7}$$

where $\lambda_v \geq 0$, $\sum_v \lambda_v = 1$ and $y_v \in \mathcal{Y}$.

(2) Define ProfSwapReg(ϕ) to equal

$$\mathrm{ProfSwapReg}(\phi) = T \cdot \left(\min_{y_v, \lambda_v} \max_{\pi^* : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v \left(u_L(\pi^*(v), y_v) - u_L(v, y_v) \right) \right), \tag{8}$$

where the outer minimum is over all valid decompositions of the form (7).

E Orthant-approachability via semi-separation

The goal of this appendix is to prove Theorem 18; that access to an efficient semi-separation oracle suffices for efficiently performing Blackwell approachability. As mentioned in Section 4, the proof of this theorem follows very closely from the proof of [Daskalakis et al., 2024a], who proved this theorem for the special class of approachability problems arising from linear swap-regret minimization, but in a fairly general way that easily extends to the setting of Theorem 18.

To avoid the unnecessary redundancy of copying the entire proof of [Daskalakis et al., 2024a] with minor changes (but still in an attempt to be relatively self-contained here), we will cite two main ingredients from [Daskalakis et al., 2024a] (describing whatever changes, if any, need to be made to the proof to extend them to our setting) and then describe how to combine them into an efficient algorithm for orthant-approachability.

The first ingredient is a procedure [Daskalakis et al., 2024a] call *Shell Projection.* Shell Projection can be thought of as a "semi-separation" variant of the operation of projecting an arbitrary u to the set $\mathcal{U}$ (a common primitive in many learning algorithms). In Shell Projection, instead of returning a projection of u onto $\mathcal{U}$ (which is computationally difficult without an actual separation oracle for $\mathcal{U}$), we instead return a projection of u onto some convex superset $\widetilde{\mathcal{U}}$ of $\mathcal{U}$, but with the guarantee that this projection is response-satisfiable.

Before we present this theorem, recall that a semi-separation oracle for a set $\mathcal{U}$ of bi-affine functions takes any bi-affine function and either returns that u is response-satisfiable or produces a hyperplane separating u from $\mathcal{U}$. Throughout the rest of this section, we will let $\mathcal{U}_{RS}$ denote the (non-convex) set of response-satisfiable bi-affine functions u (i.e., all u with the property that there exists some $x \in \mathcal{X}$ such that $u(x, y) \leq 0$ for all $y \in \mathcal{Y}$). It is worth mentioning that nothing about the Shell Projection routine below (or the one originally presented in [Daskalakis et al., 2024a]) requires anything specific about the structure of response-satisfiable bi-affine functions – indeed, the same theorem holds for a general notion of semi-separation of a convex set $\mathcal{K}$ relative to a (possibly non-convex) superset $\mathcal{K}_{NC} \supseteq \mathcal{K}$, where given a point x in the ambient space of $\mathcal{K}$, the semi-separation oracle either returns that x belongs to $\mathcal{K}_{NC}$ or returns a separating hyperplane separating x from $\mathcal{K}$.

Theorem 27 (Shell Projection). *There is an algorithm* ShellProj *which takes as input:*

- *A semi-separation oracle separating a convex set $\mathcal{U}$ relative to some set $\mathcal{U}_{RS} \supset \mathcal{U}$,*
- *A known ball of radius ρ contained within $\mathcal{U}$,*
- *A superset $\mathcal{U}'$ of $\mathcal{U}$ with* $\mathrm{diam}(\mathcal{U}') \leq D$*, provided via an efficient separation oracle,*
- *An element $u \in \mathcal{U}'$, and*
- *A precision $\varepsilon > 0$,*

runs in time $\mathrm{poly}(\dim(\mathcal{U}), \varepsilon^{-1}, \rho^{-1}, D)$, *and returns:*

- *A convex set* $\widetilde{\mathcal{U}}$ *satisfying* $\mathcal{U} \subseteq \widetilde{\mathcal{U}} \subseteq \mathcal{U}'$, *constructed by intersecting* $\mathcal{U}'$ *with at most* $\mathrm{poly}(\dim(\mathcal{U}), \varepsilon^{-1}, \rho^{-1}, D)$ *half-spaces, and*
- *A* response-satisfiable $\tilde{u} \in \widetilde{\mathcal{U}}$ *satisfying*

$$||\tilde{u} - \mathrm{Proj}_{\widetilde{\mathcal{U}}}(u)|| \leq \varepsilon.$$

Proof. The proof directly follows from the proof of Theorem 4.4 in [Daskalakis et al., 2024a], with the following changes:

- $\Phi(\mathcal{P})$, Φ_{FP}, $\tilde{\Phi}$, $\mathcal{M}$, ϕ, and ϕ' should be replaced by $\mathcal{U}$, $\mathcal{U}_{RS}$, $\widetilde{\mathcal{U}}$, $\mathcal{U}'$, u, and u' respectively.
- Theorem 4.4 uses the fact that if $\mathcal{P}$ circumscribes and is inscribed in balls of radii r and R respectively, then $\Phi(\mathcal{P})$ contains a ball of radius $r/2R$. Here, we simply make the assumption that $\mathcal{U}$ contains a ball of radius ρ (and so, ρ^{-1} appears in place of R/r in our time complexity).
- The dimension of $\Phi(\mathcal{P})$ is $d(d+1)$ (where $d = \dim(\mathcal{P})$), and a polynomial dependence in d appears in the time complexity. Instead, we directly include a polynomial dependence on $\dim(\mathcal{U})$ in our time complexity.

At a very high-level, this Shell Projection routine works by first strengthening the semi-separation oracle into a subroutine called Shell Ellipsoid, which can semi-separates *convex sets* $\mathcal{K}$ from $\mathcal{U}$ instead of individual points (returning either a hyperplane separating $\mathcal{K}$ from $\mathcal{U}$, or a point in $\mathcal{K}$ belonging to $\mathcal{U}_{RS}$). They then repeatedly run Shell Ellipsoid on larger and larger balls centered at u until they find one with a point $\tilde{u}$ in $\mathcal{U}_{RS}$ – by the guarantees of Shell Ellipsoid, they can then be guaranteed that the point $\tilde{u}$ is close to a projection of u onto some set containing $\mathcal{U}$. □

The second ingredient we will need from [Daskalakis et al., 2024a] is a slight variant of projected gradient descent they call Shell Gradient Descent (Algorithm 2 in Section 4.3 of [Daskalakis et al., 2024a]). Shell Gradient Descent solves the following variant of standard online linear optimization (that we will call Shell OLO). In Shell OLO, at the beginning of every round t, the learner is told (i.e., given efficient oracle access to) a "shell" set $\mathcal{X}_t$, from which they must pick their action x_t in round t. Their goal is to compete with the best fixed action in (some subset of) the intersection of all the sets $\mathcal{X}_t$. [Daskalakis et al., 2024a] prove the following theorem.

Theorem 28 (Shell Gradient Descent). *Let* $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_T$ *be an arbitrary sequence of convex sets satisfying* $\mathcal{X}_t \subseteq \mathcal{B}_d(0, D)$, *let* $\ell_1, \ell_2, \dots, \ell_T \in [-1, 1]^d$ *be an arbitrary sequence of adversarial losses, and let* $\mathcal{X}$ *be an arbitrary subset of* $\bigcap_{t=1}^{T} \mathcal{X}_t$. *Then, if we choose* $x_1 \in \mathcal{X}_1$ *arbitrarily and choose* $x_t = \mathrm{Proj}_{\mathcal{X}_t}(x_{t-1} - \eta_{t-1}\ell_{t-1})$ *(for some sequence of step sizes* $\eta_t > 0$*), this sequence of actions satisfies*

$$\max_{x^* \in \mathcal{X}} \sum_{t=1}^{T} \langle x_t - x^*, \ell_t \rangle \leq \frac{D^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2} ||\ell_t||^2$$

Proof. See Theorem 4.3 of [Daskalakis et al., 2024a]. (Note that this follows nearly immediately from the standard analysis of projected gradient descent). □

We now describe how these pieces fit together to produce an efficient algorithm for orthant-approachability (and thus, allow us to prove Theorem 18). The starting point is the main idea behind the standard reduction from approachability to online linear optimization (popularized by [Abernethy et al., 2011]), which is that if we can choose a sequence of bi-affine functions $u_t \in \mathcal{U}$ that forces the quantity

$$\mathrm{Reg}^{\mathrm{dual}}(\mathbf{u}, \mathbf{x}, \mathbf{y}) = \max_{u^* \in \mathcal{U}} \sum_{t=1}^{T} u^*(x_t, y_t) - \sum_{t=1}^{T} u_t(x_t, y_t)$$

to grow sublinearly in T, and in addition choose our actions $x_t \in \mathcal{X}$ so that $u_t(x_t, y) \leq 0$ for any $y \in \mathcal{Y}$, it follows that the quantity

$$\mathrm{AppLoss}(\mathbf{x}, \mathbf{y}) = \max_{u^* \in \mathcal{U}} \sum_{t=1}^{T} u^*(x_t, y_t)$$

is at most $\mathrm{Reg}^{\mathrm{dual}}(\mathbf{u}, \mathbf{x}, \mathbf{y})$ and is therefore sublinear in T.

Ordinarily, if we are given oracle access to $\mathcal{U}$, we can guarantee that $\mathrm{Reg}^{\mathrm{dual}}(\mathbf{u}, \mathbf{x}, \mathbf{y}) = o(T)$ by simply running any no-external-regret algorithm for online linear optimization where the action set is $\mathcal{U}$ and the reward function in round t is the linear function sending u to $u(x_t, y_t)$. But without oracle access to $\mathcal{U}$, it is hard to even guarantee that u_t belongs to $\mathcal{U}$. The key observation of [Daskalakis et al., 2024a] is that it is not necessary that u_t belong $\mathcal{U}$, but just that each u_t is response-satisfiable (so that $u_t(x_t, y_t) \leq 0$ holds for some $x_t \in \mathcal{X}$ independently of y_t) and that the sequence of u_t incurs low regret.

This is where Theorems 27 and 28 enter the picture. We will choose our sequence u_t by running Shell Gradient Descent over sets generated by ShellProj. This gives us the regret guarantees of Shell Gradient Descent (Theorem 28) while also ensuring that each u_t is response-satisfiable (by Theorem 27).

Algorithm 1 Orthant-approachability algorithm via semi-separation (analogue of Algorithm 4 in [Daskalakis et al., 2024a])

```
Input: a superset $\mathcal{U}'$ of $\mathcal{U}$ contained within $\mathcal{B}_d(D)$, a ball of radius $\rho$ contained within $\mathcal{U}$, and a semi-separation oracle separating $\mathcal{U}$ from $\mathcal{U}_{RS}$.
Set step size $\eta := \frac{D}{D_{\mathcal{X}} D_{\mathcal{Y}}} T^{-1/2}$ and precision $\epsilon := \frac{D}{D_{\mathcal{X}} D_{\mathcal{Y}}} T^{-1/2}$
Let $u_1$ be any point in $\mathcal{U}'$ and $x_1$ be any point in $\mathcal{X}$
for $t = 1, 2, \ldots, T$ do
    Output $x_t \in \mathcal{X}$ and receive optimizer action $y_t \in \mathcal{Y}$.
    Set $L_t \in \mathbb{R}^{\dim(\mathcal{U})}$ so that for any element $u \in \mathcal{U}$, $\langle u, L_t \rangle = -u(x_t, y_t)$.
    Run ShellProj on $u_t - \eta L_t$ with superset $\mathcal{U}'$ and precision $\epsilon$, receiving a shell set $\widetilde{\mathcal{U}}_{t+1}$ (only used in the regret analysis) and response-satisfiable element $u_{t+1} \in \widetilde{\mathcal{U}}_{t+1} \cap \mathcal{U}_{RS}$.
    Compute an $x_{t+1} \in \mathcal{X}$ such that $u_{t+1}(x_{t+1}, y) \leq 0$ for all $y \in \mathcal{Y}$ (guaranteed to exist since $u$ is response-satisfiable, and computable efficiently via linear programming).
end
```

Theorem 29 (Approachability via a semi-separation oracle (restatement of Theorem 18)). *Consider an orthant-approachability instance $(\mathcal{X}, \mathcal{Y}, \mathcal{U})$ where $\mathcal{X} \subseteq \mathcal{B}(D_{\mathcal{X}})$, $\mathcal{Y} \subseteq \mathcal{B}(D_{\mathcal{Y}})$, and*

$\mathcal{B}(u_0, \rho) \subseteq \mathcal{U} \subseteq \mathcal{B}(D)$ (for some known u_0 and radius ρ). If we have access to efficient separation oracles for $\mathcal{X}$ and $\mathcal{Y}$, and an efficient semi-separation oracle for the set $\mathcal{U}$, then Algorithm 1 has the guarantee that

$$\max_{u \in \mathcal{U}} \frac{1}{T} \sum_{t=1}^{T} u\left(x_t, y_t\right) \leq O(D_{\mathcal{X}} D_{\mathcal{Y}} D \sqrt{T}),$$

and runs in time $\mathrm{poly}(D_{\mathcal{X}}, D_{\mathcal{Y}}, D, \rho^{-1}, \dim(\mathcal{U}), T)$.

Proof. This proof closely follows the proof of Theorem 4.5 in [Daskalakis et al., 2024a].

Note that the sequence u_t generated by Algorithm 1 is almost the same sequence as would be generated by running Shell Gradient Descent on the sequence of shell sets $\widetilde{\mathcal{U}}_t$, with the only caveat that u_{t+1} is distance at most ε from the true projection of $u_t - \eta L_t$ onto $\widetilde{\mathcal{U}}_{t+1}$. The guarantees of Theorem 28 then imply that

$$\max_{u^* \in \mathcal{U}} \sum_{t=1}^{T} \langle u_t - u^*, L_t \rangle \leq \frac{D^2}{2\eta} + \sum_{t=1}^{T} \frac{\eta}{2} ||L_t||^2 + \varepsilon \sum_{t=1}^{T} ||L_t|| = O(D D_{\mathcal{X}} D_{\mathcal{Y}} \sqrt{T}), \tag{9}$$

where here we have used the fact that $||L_t|| \leq D_{\mathcal{X}} D_{\mathcal{Y}}$ and $\eta = \varepsilon = \frac{D}{D_{\mathcal{X}} D_{\mathcal{Y}}} T^{-1/2}$. But since $\langle u_t, L_t \rangle = -u_t(x_t, y_t)$, the left hand side of (9) is equal to $\max_{u^* \in \mathcal{U}} \sum_{t=1}^{T} u^*(x_t, y_t) - \sum_{t=1}^{T} u_t(x_t, y_t) = \mathrm{Reg}^{\mathrm{dual}}(\mathbf{u}, \mathbf{x}, \mathbf{y})$. Moreover, since each $u_t(x_t, y_t) \leq 0$ (by the choice of x_t), we have that

$$\max_{u \in \mathcal{U}} \frac{1}{T} \sum_{t=1}^{T} u\left(x_t, y_t\right) \leq \mathrm{Reg}^{\mathrm{dual}}(\mathbf{u}, \mathbf{x}, \mathbf{y}) \leq O(D D_{\mathcal{X}} D_{\mathcal{Y}} \sqrt{T}),$$

as desired. □

F Additional results on game-agnostic learning

F.1 Profile swap regret is not computable from the game agnostic transcript

In this appendix, we show that it is not possible to compute the profile swap regret $\mathrm{ProfSwapReg}(\mathbf{x}, \mathbf{y})$ from the game-agnostic transcript $(\mathbf{x}, \mathbf{r})$ corresponding to the game-dependent transcript $(\mathbf{x}, \mathbf{y})$. Specifically, we have the following theorem.

Theorem 30. *There exists a polytope game G, and two transcripts $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ that have the same game-agnostic transcript $(\mathbf{x}, \mathbf{r})$, but where $\mathrm{ProfSwapReg}(\mathbf{x}_1, \mathbf{y}_1) \neq \mathrm{ProfSwapReg}(\mathbf{x}_2, \mathbf{y}_2)$.*

Proof. Our polytope game G will be a variant on the polytope game used to separate profile swap regret and polytope swap regret in the proof of Theorem 8. Specifically, we will consider the game where $\mathcal{X} = \Delta_2^2$ and $\mathcal{Y} = \Delta_2^2 \times \Delta_4$. We will write elements $y \in \mathcal{Y}$ in the form $y = (y_r, y_s)$ with $y_r \in \Delta_2^2$ and $y_s \in \Delta_4$, and write $\mathrm{Proj}_1(y_r, y_s) = y_r \in \Delta_2^2$ and $\mathrm{Proj}_2(y_r, y_s) = y_s \in \Delta_4$. Finally, we define $u_L(x, y) = \langle x, \mathrm{Proj}_1(y) \rangle$.

As in the proof of Theorem 8, we will write $\mathcal{V}(\mathcal{X}) = \{v_{11}, v_{12}, v_{21}, v_{22}\}$, where $v_{11} = (1, 0, 1, 0)$, $v_{12} = (1, 0, 0, 1)$, $v_{21} = (0, 1, 1, 0)$, and $v_{22} = (0, 1, 0, 1)$. We will likewise write $\mathcal{V}(\mathcal{Y}) = \{w_{ijk}\}_{i,j \in [2], k \in [4]}$, where $w_{ijk} = (v_{ij}, e_k)$. In particular, note that $\mathrm{Proj}_1(w_{ijk}) = v_{ij}$.

We now define two transcripts of play. Each transcript of play will be divided into four epochs of $T/4$ rounds during which both players play a fixed action. In the first transcript $(\mathbf{x}_1, \mathbf{y}_1)$, these four action pairs are $((v_{11}, w_{111}), (v_{12}, w_{121}), (v_{21}, w_{211}), (v_{22}, w_{111}))$. In the second transcript $(\mathbf{x}_2, \mathbf{y}_2)$, these four action pairs are $((v_{11}, w_{111}), (v_{12}, w_{122}), (v_{21}, w_{213}), (v_{22}, w_{114}))$. Note that since $\mathbf{x}_{1,t} = \mathbf{x}_{2,t}$ and $\mathrm{Proj}_1(\mathbf{y}_{1,t}) = \mathrm{Proj}_1(\mathbf{y}_{2,t})$ for each $t \in [T]$, it follows that both transcripts have the same game-agnostic transcript $(\mathbf{x}, \mathbf{r})$.

Let ϕ_1 be the CSP corresponding to the first transcript and ϕ_2 be the CSP corresponding to the second transcript. Note that by following the same derivation as in the proof of Theorem 8, we can write

$$\phi_1 = \frac{1}{4}(v_{12} \otimes (w_{121} + w_{111}) + v_{21} \otimes (w_{211} + w_{111})),$$

from which we can conclude that $\mathsf{ProfSwapReg}(\phi_1) = 0$.

On the other hand, we claim that the only decomposition of ϕ_2 into the form

$$\phi_2 = \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v (v \otimes y_v),. \tag{10}$$

is the decomposition given by its construction as a CSP, namely,

$$\phi_2 = \frac{1}{4}\left(v_{11} \otimes w_{111} + v_{12} \otimes w_{122} + v_{21} \otimes w_{213} + v_{22} \otimes w_{114}\right). \tag{11}$$

To see this, for any $k \in [4]$, consider the bilinear function ρ_k defined via $\rho_k(x \otimes y) = \langle \mathrm{Proj}_2(y), e_k \rangle x$. For each k, there is exactly one $v_k \in \mathcal{V}(\mathcal{X})$ and $w_k \in \mathcal{V}(\mathcal{Y})$ with the property that $v_k \otimes w_k$ is the only term in (11) that does not vanish under ρ_k (e.g., for $k = 4$ it is $v_{22} \otimes w_{114}$). But applying ρ_k to (10),

$$\rho_k(\phi_2) = \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v \langle y_v, e_k \rangle v.$$

Since v_k is an extreme point of $\mathcal{X}$, this implies that $\langle y_v, e_k \rangle$ can be non-zero only for $v = v_k$ (for which value it must equal 1). It follows that the only possible decomposition of ϕ_2 is that given in (11). But then, since $v_{22} \notin \mathsf{BR}_L(w_{114}) = \{v_{11}\}$, it follows that $\mathsf{ProfSwapReg}(\phi_2) > 0$ and therefore $\mathsf{ProfSwapReg}(\phi_1) \neq \mathsf{ProfSwapReg}(\phi_2)$.

□

G Additional results on correlated equilibria

G.1 Mediator protocols correspond to normal-form CE

In this appendix we make clear the connection between the incentive-compatible signaling schemes a mediator can implement third-party mediator and normal-form correlated equilibria. In particular, we show that there is no need for a mediator to ever send recommendations to the two players that are not extreme points of $\mathcal{V}(\mathcal{X})$ or $\mathcal{V}(\mathcal{Y})$, and hence that the set of valid mediator outcomes can be exactly characterized by normal-form CE.

Specifically, for any (finite support) distribution $\sigma \in \Delta(\mathcal{X} \times \mathcal{Y})$, we can consider the signaling scheme where a mediator samples a strategy profile $(x, y) \sim \sigma$, privately recommends playing x to the $\mathcal{X}$-player, and privately recommends playing y to the $\mathcal{Y}$-player. We say this signaling scheme is incentive compatible if neither player has incentive to deviate given their observation. That is, for any $x_{\text{sig}} \in \mathcal{X}$, there should not exist an $x' \in \mathcal{X}$ such that:

$$\mathop{\mathbb{E}}_{(x,y)\sim\sigma}[u_X(x', y) \mid x = x_{\text{sig}}] > \mathop{\mathbb{E}}_{(x,y)\sim\sigma}[u_X(x, y) \mid x = x_{\text{sig}}].$$

(i.e., the $\mathcal{X}$-player cannot increase their utility by playing x' every time they receive the signal x_{sig}). Similarly, for any $y_{\text{sig}} \in \mathcal{Y}$, there should not exist a $y' \in \mathcal{Y}$ such that:

$$\mathop{\mathbb{E}}_{(x,y)\sim\sigma}[u_Y(x, y') \mid y = y_{\text{sig}}] > \mathop{\mathbb{E}}_{(x,y)\sim\sigma}[u_Y(x, y) \mid y = y_{\text{sig}}].$$

The following lemma shows that we can always convert an incentive-compatible signaling scheme supported on the entire set $\mathcal{X} \times \mathcal{Y}$ to one supported on pairs of pure strategies $\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y})$ with the same utility for both players (in fact, that induce the same CSP).

Lemma 31. *Let $\sigma \in \Delta(\mathcal{X} \times \mathcal{Y})$ be an incentive compatible signaling scheme. Let $\sigma' \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ be the signaling scheme formed by first sampling a strategy profile (x, y) from σ, decomposing x and y into convex combinations of pure strategies $x = \sum_{v\in\mathcal{V}(\mathcal{X})} \lambda_v v$ and $y = \sum_{w\in\mathcal{V}(\mathcal{Y})} \mu_w w$, and then returning the pure strategy profile (v, w) with probability $\lambda_v \mu_w$. Then σ' is an incentive compatible signaling scheme, and $\mathbb{E}_{(x,y)\sim\sigma}[x \otimes y] = \mathbb{E}_{(v,w)\sim\sigma'}[v \otimes w]$.*

Proof. The fact that $\mathbb{E}_{(x,y)\sim\sigma}[x \otimes y] = \mathbb{E}_{(v,w)\sim\sigma'}[v \otimes w]$ follows directly from the construction of σ' (in particular, $\sum_{v,w} \lambda_v \mu_w (v \otimes w) = (\sum_v \lambda_v v) \otimes (\sum_w \mu_w w) = x \otimes y$). It therefore suffices to show that σ' is incentive compatible.

We will do this for the $\mathcal{X}$-player (it follows for the $\mathcal{Y}$-player by symmetry). Consider any $x \in \mathcal{X}$ which is recommended to the $\mathcal{X}$-player with positive probability under σ. Let $y(x) \in \mathcal{Y}$ be the expected action recommended to the $\mathcal{Y}$-player, conditioned on the fact that x is recommended to the $\mathcal{X}$-player. Note that since σ is incentive compatible, this means that it must be the case that $x \in \mathsf{BR}_X(y(x))$ (or the $\mathcal{X}$ player could increase their utility by deviating from x to any element of $\mathsf{BR}_X(y(x))$).

Now, consider any $v^* \in \mathcal{V}(\mathcal{X})$ such that $\lambda_{v^*} > 0$ in some decomposition of x of the form $x = \sum_{v\in\mathcal{V}(\mathcal{X})} \lambda_v v$. We claim that it must also be the case that $v^* \in \mathsf{BR}_X(y(x))$. If not, the $\mathcal{X}$-player could again improve their utility by deviating from x to the strategy formed by replacing the $\lambda_{v^*} v^*$ term in the decomposition $\sum_{v\in\mathcal{V}(\mathcal{X})} \lambda_v v$ of x with $\lambda_{v^*} v'$ for some $v' \in \mathsf{BR}_X(y(x))$. But now, if we let $y(v^*) \in \mathcal{Y}$ to be the expected message in σ' received by the $\mathcal{Y}$-player conditioned on the $\mathcal{X}$-player receiving v^*, $y(v^*)$ is a convex combination of elements of the form $y(x)$ for x with $\lambda_{v^*} > 0$. It follows that $v^* \in \mathsf{BR}_X(y(v^*))$, and therefore that σ' is incentive-compatible. □

G.2 Every profile CE can be reached by no-profile-swap-regret dynamics

Lemma 32. *Given any profile CE ϕ, there exist profile swap regret minimizing for the two players that converge to this CSP when employed against each other.*

Proof. Let $\phi = \sum_{k=1}^{K} \lambda_k (x_{(k)} \otimes y_{(k)})$ be a decomposition of the CSP ϕ with $K \leq d_{\mathcal{X}}$ (where $\mathcal{X} \subseteq \mathbb{R}^{d_{\mathcal{X}}}$) – Lemma 26 guarantees such a decomposition. In case any of the λ_k's are not rational – we associate it with a sequence $\lambda_k^1, \lambda_k^2 \cdots$ of positive rational numbers converging to λ_k. The two algorithms attempt to follow the schedule described below with any deviation observed triggering a default to a standard profile swap regret minimizing algorithm (such as the algorithm guaranteed by Theorem 14). The schedule consists of an infinite series of epochs, the i-th epoch consists of $\sum_{k=1}^{K} \lambda_k^i$ rounds – in each of the first λ_1^i rounds in this epoch, the two players play x_1 and y_1 respectively; in the next λ_2^i rounds in this epoch, the two players play x_2 and y_2 respectively and so on. It it not hard to see that both players following the schedule leads to convergence to the CSP ϕ, which is a profile CE. On the other hand, any deviation from the schedule by an opposing player results in a reset to a profile swap regret minimizing algorithm, so both player's algorithms retain the worst case profile swap regret minimizing property. □

G.3 One-sided mediator protocols and profile swap regret

Lemma 33. *Let $\phi \in \mathcal{X} \otimes \mathcal{Y}$ be a CSP with the property that* $\mathsf{ProfSwapReg}_{\mathcal{X}}(\phi) = 0$. *Then there exists a vertex-game CSP $\phi^V \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ with the property that* $\mathsf{NFSwapReg}(\phi^V) = 0$ *and* $\mathsf{Proj}(\phi^V) = \phi$.

Proof. Consider a normal form swap-regret minimizing algorithm $\mathcal{A}$ in the vertex-game. We define the associated menu $\mathcal{M}^V \subseteq \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$. Recall the operation $\mathsf{Proj} : \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y})) \rightarrow \mathcal{X} \otimes \mathcal{Y}$ and define $\mathsf{Proj}(\mathcal{M}^V) = \{\mathsf{Proj}(\psi^V) | \forall \psi^V \in \mathcal{M}^V\}$. We show that $\mathsf{Proj}(\mathcal{M}^V) = \mathcal{M}_{NPSR}$ to prove this lemma. Consider any transcript of play of $\mathcal{A}$ with CSP $\psi^V \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ - with actions x_t^V, y_t^V in the t-th round. This transcript must have zero normal form swap-regret, and consequently the transcript with actions $\mathbb{E}[x_t^V], \mathbb{E}[y_t^V]$ and CSP $\psi \in \mathcal{X} \otimes \mathcal{Y}$ in the t-th round must have zero polytope and profile swap regret (by Theorems 2 and 8). This implies $\mathsf{Proj}(\mathcal{M}^V) \subseteq \mathcal{M}_{NPSR}$. On the other hand, we know that $\mathcal{M}^V$ is a no-swap-regret menu in a normal-form game and therefore we know it is a polytope with vertices of the form $x_v^V \otimes y_v^V$ where x_v^V is a one-hot vector with 1 at the index $v \in \mathcal{V}(\mathcal{X})$ and y_v^V being a distribution over $\mathcal{V}(\mathcal{Y})$ (Using Lemma 9 and the observation that all forms of swap-regret are equivalent in normal form games) with $v = BR_L(y_v)$ where $\mathbb{E}[y_v^V] = y_v \in \mathcal{Y}$. Notice that $\mathbb{E}[x_v^V] = v$. Thus, the projection of this extreme point is $v \otimes y_v$, which by Lemma 9 is a vertex of $\mathcal{M}_{NPSR}$, implying $\mathsf{Proj}(\mathcal{M}^V) \supseteq \mathcal{M}_{NPSR}$ and completing the proof. □

One consequence of Lemma 33 is that the no-profile-swap-regret menu is a projection of the no-swap-regret menu of the vertex game: that is, $\mathcal{M}_{NPSR} = \mathsf{Proj}(\mathcal{M}_{NSR}^V)$, where $\mathcal{M}_{NSR}^V$ is the set of vertex-game CSPs ϕ^V with $\mathsf{NFSwapReg}(\phi^V) = 0$.

G.4 Computing profile correlated equilibria

In this appendix, we apply our efficient algorithm for minimizing profile swap regret to efficiently compute a profile correlated equilibria. Formally, we define an *ε-approximate profile CE ϕ* to be a CSP satisfying $\mathsf{ProfSwapDist}(\phi) \leq \varepsilon$ for both players.

Lemma 34. *There exists an algorithm that runs in polynomial time in $d_L, d_O, \frac{1}{\varepsilon}$ and computes an ε-approximate profile CE.*

Proof. Our algorithm simulates repeated game play between the two players, with both players employing Algorithm 1 with their respective utility functions/action spaces for $T = \Theta(\varepsilon^{-2})$ rounds. The algorithm runs in polynomial time and by Theorem 20, directly guarantees that

$\mathsf{ProfSwapDist}(\phi) = O(\sqrt{T}) \leq 1/\varepsilon$ (for an appropriate choice of the constant in $\Theta(\varepsilon^{-2})$) for both players. □

H Omitted proofs

H.1 Proof of Theorem 1

Proof of Theorem 1. When $\mathcal{X} = \Delta_m$, the set of affine linear transformations that sends $\mathcal{X}$ to itself can be expressed the set of m-by-m row-stochastic matrices (this follows since every unit vector must map to an element of Δ_m). This is a convex set whose extreme points are given by row-stochastic matrices which correspond to swap functions; i.e., 0/1 row-stochastic matrices where each row is a unit vector. Maximizing over all transformations in this set is equivalent to maximizing over all swap functions, so $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{SwapReg}(\mathbf{x}, \mathbf{y})$.

For polytope swap regret, note that when $\mathcal{X} = \Delta_m$, there is a unique way to decompose an element in $x \in \mathcal{X}$ into an element $x^V \in \Delta(\mathcal{V}(\mathcal{X}))$ (specifically, x itself provides the unique such decomposition). This means that we can simplify the definition of polytope swap regret in this case to $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \max_{\pi:\mathcal{V}(\mathcal{X})\to\mathcal{V}(\mathcal{X})}(\sum_{t=1}^{T} u_L(\pi(x_t), y_t) - u_L(x_t, y_t))$, which is identical to the definition of $\mathsf{SwapReg}(\mathbf{x}, \mathbf{y})$.

Finally, again, the fact that any action $x^V \in \Delta(\mathcal{V}(\mathcal{X}))$ with $\mathbb{E}_{v\sim x^V}[v] = x$ must in fact satisfy $x_i^V = x_i$ for all $i \in [m]$ means that for normal-form games, $\mathsf{NFSwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{SwapReg}(\mathbf{x}, \mathbf{y})$. □

H.2 Proof of Theorem 2

Proof of Theorem 2. We begin by proving the two inequalities above. To show that $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$, first note that we can relax the set of swap functions π we consider in the definition of polytope swap regret to all functions π from $\mathcal{V}(\mathcal{X})$ to $\mathcal{X}$ (instead of just functions from $\mathcal{V}(\mathcal{X})$ to $\mathcal{V}(\mathcal{X})$). This relaxation does not change the value of polytope swap regret, since for any vertex $v \in \mathcal{V}(\mathcal{X})$, the value of $\pi(v)$ that maximizes polytope swap regret will be achieved at an extreme point of $\mathcal{V}(\mathcal{X})$. But now, for any affine linear function $\psi : \mathcal{X} \to \mathcal{X}$, if we consider the swap function π with $\pi(v) = \psi(v)$ for all $v \in \mathcal{X}$, it is the case that $\overline{\pi}(x_t^V) = \mathbb{E}_{v\sim x_t^V}[\psi(v)] = \psi(\mathbb{E}_{v\sim x_t^V}[v]) = \psi(x_t)$ (in particular, since ψ is linear, it commutes with taking expectations). It follows that the polytope swap regret is at least $\sum_{t=1}^{T} u_L(\psi(x_t), y_t) - \sum_{t=1}^{T} u_L(x_t, y_t)$, for any linear function ψ that maps $\mathcal{X}$ to itself. Since $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y})$ is the maximum of this quantity over all such linear functions ψ, it follows that $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$.

To show that $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V)$, note that we can write $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$ as the minimum of $\mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V)$ over all choices of $\mathbf{x}^V$ and $\mathbf{y}^V$ with $x_t = \mathbb{E}[\mathbf{x}_t^V]$ and $y_t = \mathbb{E}[\mathbf{y}_t^V]$. Since the $\mathbf{x}^V$ and $\mathbf{y}^V$ in this theorem also satisfy these constraints, it immediately follows that $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V)$.

We now provide examples of transcripts exhibiting the gaps described in the second part of the theorem. An example of a family of transcripts where $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = 0$ but $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \Omega(T)$ can be found in Theorem 7 of [Mansour et al., 2022].

To construct a family of transcripts where $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = 0$ but $\mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V) = \Omega(T)$, consider the polytope game where $\mathcal{X} = [0, 1]^2$ (and therefore $V(\mathcal{X}) = \{(0, 0), (1, 0), (1, 1), (0, 1)\}$) and $\mathcal{Y} = [-1, 1]$, with $u_L((x_1, x_2), y) = y(x_1 - x_2)$. Consider the transcript $(\mathbf{x}, \mathbf{y})$ where $x_t = (1/2, 1/2)$ for all $t \in [T]$ but $y_t = -1$ for rounds $t \in [1, T/2]$ and $y_t = 1$ for rounds $t \in [T/2 + 1, T]$.

Note that if we decompose every x_t into the distribution $\frac{1}{2}(0,0) + \frac{1}{2}(1,1) \in \Delta(V(\mathcal{X}))$, there is no swap function $\pi^* : V(\mathcal{X}) \to V(\mathcal{X})$ that will improve the utility of the learner – for example, on average when the learner plays the vertex $(0,0)$, they face the average optimizer strategy of $\overline{y} = 0$, and all elements of $V(\mathcal{X})$ are best responses to $\overline{y}$. It follows that $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = 0$.

On the other hand, consider the vertex game transcript where $x_t^V = \frac{1}{2}(0,0) + \frac{1}{2}(1,1)$ for rounds $t \in [1, T/2]$, and $x_t^V = \frac{1}{2}(0,1) + \frac{1}{2}(1,0)$ for rounds $t \in [T/2+1, T]$ (with $y_t^V = y_t$ for all $t \in [T]$). It is straightforward to check that this transcript satisfies $\mathbb{E}[x_t^V] = x_t$ and $\mathbb{E}[y_t^V] = y_t$. However, on the rounds where the learner plays $(0,0)$, they face an average optimizer strategy of $\overline{y} = -1$. Against this optimizer strategy, the unique best-response for the learner is the strategy $(0,1)$, and the learner can increase their total utility by $T/4$ by applying this swap. It follows that $\mathsf{NFSwapReg}(\mathbf{x}^V, \mathbf{y}^V) = \Omega(T)$. □

H.3 Proof of Theorem 3

Proof of Theorem 3. See the proof of Theorem 3.3 in [Arunachaleswaran et al., 2024] (this proof is for a normal-form game G, but does not use anything specific about the space of CSPs, and applies essentially as is to our setting).

It is worth noting that one direction of this argument – that $\mathcal{M}$ needs to contain some CSP of the form $x \otimes y$ for each $y \in \mathcal{Y}$ follows simply from the fact that the optimizer can always play the action $y_t = y$, in which case some element of this form is guaranteed to exist in $\mathcal{M}$. It is also possible to prove a slightly weaker version of the converse directly from Blackwell's approachability theorem (which is all we need in this paper): that if $\mathcal{M}$ contains some CSP of the form $x \otimes y$ for each $y \in \mathcal{Y}$, then some *subset* $\mathcal{M}'$ of $\mathcal{M}$ is a valid asymptotic menu (see Lemma 3.4 in [Arunachaleswaran et al., 2024]). Proving the full converse requires showing that we can always extend a menu $\mathcal{M}'$ to any superset of $\mathcal{M}'$ (which can be accomplished by offering the optimizer the choice to play a choice of CSPs outside $\mathcal{M}'$ – see Lemma 3.5 in [Arunachaleswaran et al., 2024]). □

H.4 Proof of Theorem 8

Proof of Theorem 8. We begin by proving the two inequalities. To show that $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y})$, let ϕ be the average CSP of the transcript $(\mathbf{x}, \mathbf{y})$, and let $\phi = \sum_k \lambda_k (x_{(k)} \otimes y_{(k)})$ be any decomposition of ϕ in the form (4). Consider also any linear transformation ψ that sends $\mathcal{X}$ to $\mathcal{X}$. Note that $\sum_k \lambda_k \operatorname{Reg}(x_{(k)}, y_{(k)}) \geq \sum_k \lambda_k (u_L(\psi(x_{(k)}), y_{(k)}) - u_L(x_{(k)}, y_{(k)})) = \frac{1}{T} \sum_t (u_L(\psi(x_t), y_t) - u_L(x_t, y_t))$, where the last equality follows by the linearity of ψ. Taking the minimum of the left-hand side of this inequality (over all decompositions of ϕ) and the maximum of the right-hand side (over all linear endomorphisms ψ), we obtain that $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y})$.

To show that $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$, note that we can rewrite $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$ in the form:

$$\begin{aligned}\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) &= \min_{\mathbf{x}^V} \max_{\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \left(\sum_{t=1}^{T} u_L(\overline{\pi}(x_t^V), y_t) - u_L(x_t, y_t) \right) \\ &= T \cdot \min_{\mathbf{x}^V} \max_{\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \left(\sum_{v \in \mathcal{X}} \alpha_v \left(u_L(\overline{\pi}(v), y_v) - u_L(v, y_v) \right) \right) \qquad (12)\end{aligned}$$

where the α_v and y_v in (12) are obtained by expanding x_t^V out as an explicit convex combination of vertices in $\mathcal{V}(\mathcal{X})$. In particular, every choice of per-round decompositions $\mathbf{x}^V$ in the outer minimum gives rise to a collection of α_v and y_v satisfying $\alpha_v \geq 0$, $\sum_v \alpha_v = 1$, and $y_v \in \mathcal{Y}$. Since profile swap regret is the minimum over *all* such decompositions (see (8)), $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) \leq \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$.

We now prove the second part of the theorem, presenting families of transcripts that exhibit the above gaps. We provide an indirect proof that there exists an example where $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = 0$ but $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) = \Omega(T)$, by noticing that Theorem 8 of [Mansour et al., 2022] provides an example of a transcript $(\mathbf{x}, \mathbf{y})$ in a polytope game with $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y}) = 0$ but that is manipulable. By Theorem 10, it must be the case that $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) > 0$. It is also possible to verify this example directly, but is computationally somewhat messy.

We now present an example where $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) = 0$ but $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \Omega(T)$. This example is adapted from a counterexample in [Rubinstein and Zhao, 2024] used to show that there exist transcripts with high polytope swap regret that were not manipulable. We will set $\mathcal{X} = \mathcal{Y} = \Delta_2^2 = \{(a_1, a_2, b_1, b_2) \mid a_1 + a_2 = 1, b_1 + b_2 = 1, a_1, a_2, b_1, b_2 \in \mathbb{R}_{\geq 0}\}$ (this can be interpreted as a Bayesian game with two types and two actions for both players). For this set, we can write $\mathcal{V}(\mathcal{X}) = \mathcal{V}(\mathcal{Y}) = \{v_{11}, v_{12}, v_{21}, v_{22}\}$, where $v_{11} = (1, 0, 1, 0)$, $v_{12} = (1, 0, 0, 1)$, $v_{21} = (0, 1, 1, 0)$, and $v_{22} = (0, 1, 0, 1)$. We will let the learner's payoff be simply $u_L(x, y) = \langle x, y \rangle$.

Now, consider the following transcript of play:

- For $0 < t \leq T/4$, $x_t = y_t = v_{11}$.
- For $T/4 < t \leq T/2$, $x_t = y_t = v_{12}$.
- For $T/2 < t \leq 3T/4$, $x_t = y_t = v_{21}$.
- For $3T/4 < t \leq T$, $x_t = v_{22}$ and $y_t = v_{11}$. **Note that $x_t \neq y_t$ for this period.**

We claim that $\mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y}) = \Omega(T)$ for this transcript of play. Indeed, note that since the learner plays an extreme point of $\mathcal{X}$ every round, any decomposition $\mathbf{x}^V$ must satisfy $x_t^V = x_t$. Now, the map π which sends v_{22} to v_{11} (and fixes all other vertices) improves the learner's utility by $T/4 = \Omega(T)$.

On the other hand, we claim that $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) = 0$. To see this, note that the CSP corresponding to this transcript can be written in the form

$$\begin{aligned}
\phi &= \frac{1}{4}\left(v_{11} \otimes v_{11} + v_{12} \otimes v_{12} + v_{21} \otimes v_{21} + v_{22} \otimes v_{11}\right) \\
&= \frac{1}{4}\left((v_{11} + v_{22}) \otimes v_{11} + v_{12} \otimes v_{12} + v_{21} \otimes v_{21}\right) \\
&= \frac{1}{4}\left((v_{12} + v_{21}) \otimes v_{11} + v_{12} \otimes v_{12} + v_{21} \otimes v_{21}\right) \\
&= \frac{1}{4}\left(v_{12} \otimes (v_{12} + v_{11}) + v_{21} \otimes (v_{21} + v_{11})\right).
\end{aligned}$$

Here, in the third line, we have used the fact that $v_{11} + v_{22} = v_{12} + v_{21}$. But now, note that $v_{12} \in \mathsf{BR}_L(v_{12} + v_{11})$, and $v_{21} \in \mathsf{BR}_L(v_{21} + v_{11})$. It follows that $\mathsf{ProfSwapReg}(\phi) = 0$, as desired. □

H.5 Proof of Lemma 26

Proof of Lemma 26. Fix any $k \in [K]$, and decompose $x_{(k)}$ arbitrarily into a convex combination $\sum_{s=1}^{S_k} \gamma_{s,k} v_{s,k}$ of vertices $v_{s,k} \in \mathcal{V}(\mathcal{X})$. Note that since the function $\mathrm{Reg}(x, y)$ is convex in x for any fixed y (it is the maximum of a collection of linear functions), it follows that $\sum_{s=1}^{S_k} \gamma_{s,k} \mathrm{Reg}(v_{s,k}, y_{(k)}) \leq \mathrm{Reg}(x_{(k)}, y_{(k)})$. It follows that the decomposition $(\lambda'_k, x'_{(k)}, y'_{(k)})$ formed by decomposing each $x_{(k)}$ in this way and aggregating the terms with the same vertex v has the desired property. □

H.6 Proof of Lemma 9

Proof of Lemma 9. This follows almost directly from the definition of profile swap regret. Note that if $\mathrm{ProfSwapReg}(\phi) = 0$, then this implies that we can write it in the form $\phi = \sum \lambda_k (x'_k \otimes y'_k)$ where each pair (x'_k, y'_k) satisfies $\mathrm{Reg}(x'_k, y'_k) = 0$, and therefore that $x'_k \in \mathrm{BR}_L(y'_k)$ (and so we have expressed ϕ as a convex combination of such points). Conversely, if $x \in \mathrm{BR}_L(y)$, then $\mathrm{ProfSwapReg}(x \otimes y) = 0$ (since $\mathrm{Reg}(x, y) = 0$). □

H.7 Proof of Theorem 10

Proof of Theorem 10. We will first show that the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$ is non-manipulable (therefore implying that any no-profile-swap-regret algorithm is non-manipulable). It suffices to show that, for any optimizer payoff u_O and any CSP $\phi \in \mathcal{M}_{NPSR}$, $u_O(\phi) \leq \mathrm{Stack}(G, u_O)$. But since $\mathrm{Stack}(G, u_O) = V_O(\mathcal{M}_{NPSR}, u_O)$ (by Lemma 9), this is immediately true.

Conversely, if $\mathcal{A}$ is not a no-profile-swap-regret algorithm, then $\mathcal{M}(\mathcal{A})$ must contain a CSP $\phi \notin \mathcal{M}_{NPSR}$. By the separating hyperplane theorem, there must exist a linear payoff function u_O such that $u_O(\phi) > 0$ but $u_O(\phi') \leq 0$ for all $\phi' \in \mathcal{M}_{NPSR}$. But if $u_O(\phi') \leq 0$ for all $\phi' \in \mathcal{M}_{NPSR}$, then we must also have $\mathrm{Stack}(G, u_O) \leq 0$ (by Lemma 9). Therefore $V_O(\mathcal{M}(\mathcal{A}), u_O) \geq u_O(\phi) > \mathrm{Stack}(G, u_O)$, and $\mathcal{M}(\mathcal{A})$ is therefore manipulable.

Finally, we can quantify the degree of manipulation in the above argument as follows. If $\mathrm{ProfSwapDist}(\mathcal{A}^T) \leq R(T)$, then for any $\phi \in \mathcal{M}(\mathcal{A}^T)$, there exists a $\phi' \in \mathcal{M}_{NPSR}$ such that $||\phi - \phi'||_2 \leq R(T)/T$. In particular, for any optimizer payoff u_O, we have that $u_O(\phi) - \mathrm{Stack}(G, u_O) \leq u_O(\phi) - u_O(\phi') \leq ||u_O||_2 \cdot ||\phi - \phi'||_2 \leq \sqrt{d_L d_O}(R(T)/T)$ (where the last inequality follows from the fact that $||u_O||_2 \leq \sqrt{d_L d_O}$ if $||u_O||_\infty \leq 1$).

Likewise, if $\mathrm{ProfSwapDist}(\mathcal{A}^T) \geq R(T)$, then there is a $\phi \in \mathcal{M}$ such that $||\phi - \phi'||_2 \geq R(T)/T$ for any $\phi' \in \mathcal{M}_{NPSR}$. By the separating hyperplane theorem there then exists a u_O with unit $||u_O||_2 = 1$ such that $u_O(\phi) - \max_{\phi' \in \mathcal{M}_{NPSR}} u_O(\phi') \geq R(T)/T$. Since $\mathrm{Stack}(G, u_O) = \max_{\phi' \in \mathcal{M}_{NPSR}} u_O(\phi')$, this implies that the menu $\mathcal{M}$ is at least $(R(T)/T)$-manipulable. □

H.8 Proof of Theorem 11

Proof of Theorem 11. We will begin by showing the second part of this theorem: that any no-(external)-regret algorithm $\mathcal{A}$ for G has the property that $\mathcal{M}_{NPSR} \subseteq \mathcal{M}(\mathcal{A})$.

Recall that $\mathcal{M}_{NPSR}$ is the convex hull of all points of the form $\mathrm{BR}_L(y) \otimes y$. B We will show that every point of this form is in $\mathcal{M}(\mathcal{A})$. For every $\hat{x} \otimes \hat{y}$ pair such that $\hat{x} = \mathrm{BR}_L(\hat{y})$, there are two possibilities:

- $\hat{x}$ is the unique best response to $\hat{y}$. Assume for contradiction that $\hat{x} \otimes \hat{y}$ is not in $\mathcal{M}(\mathcal{A})$. By Theorem 3, there must exist at least one point of the form $x \otimes \hat{y}$ in $\mathcal{M}(\mathcal{A})$. Thus, there exists a point of the form $x \otimes \hat{y}$ such that x is not a best response to $\hat{y}$. But this point incurs external regret, and therefore this is a contradiction.
- $\hat{x}$ is not the unique best response to $\hat{y}$. Then, note that by our non-degeneracy assumption on u_L, $\hat{x}$ is not weakly dominated. Thus there must exist a $y^* \in \mathcal{Y}$ that uniquely incentivizes $\hat{x}$. Therefore, all points of the form $y(\alpha) = \alpha y^* + (1-\alpha)\hat{y}$ uniquely incentivize $\hat{x}$ for $\alpha \in [0, 1]$. Therefore, all points $\hat{x} \otimes y(\alpha)$ are contained within $\mathcal{M}(\mathcal{A})$. As $\mathcal{M}(\mathcal{A})$ is closed, taking the limit of $\alpha \longrightarrow 0$ implies that $\hat{x} \otimes y(0) = \hat{x} \otimes \hat{y} \in \mathcal{M}(\mathcal{A})$.

To establish the first part of this theorem, note that by the definition of $\mathcal{M}_{NPSR}$, if $\mathcal{A}$ is a no-profile-swap-regret algorithm, then $\mathcal{M}(\mathcal{A}) \subseteq \mathcal{M}_{NPSR}$. On the other hand, since no-profile-swap-regret implies no-external-regret, by the above argument we have that $\mathcal{M}_{NPSR} \subseteq \mathcal{M}(\mathcal{A})$. Therefore, $\mathcal{M}_{NPSR} = \mathcal{M}(\mathcal{A})$. □

H.9 Proof of Theorem 12

Give a menu $\mathcal{M}$, let $\mathcal{M}^+$ represent the set of all extreme points with maximum u_L value over $\mathcal{M}$. Furthermore, let $\mathcal{M}^-$ represent the set of all extreme points with minimum u_L value over $\mathcal{M}$.

This proof uses Lemma 4.4 from [Arunachaleswaran et al., 2024]. While [Arunachaleswaran et al., 2024] is primarily about normal-form games in which the strategy polytopes are simplices, their proof of this Lemma makes no such assumptions.

Lemma 35 (Lemma 4.4 in [Arunachaleswaran et al., 2024]). *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two distinct asymptotic menus where $\mathcal{M}_1^+ = \mathcal{M}_2^+$. If $\mathcal{M}_2 \setminus \mathcal{M}_1 \neq \emptyset$, then there exists a u_O for which $V_L(\mathcal{M}_1, u_O) > V_L(\mathcal{M}_2, u_O)$.*

Proof of Theorem 12. Consider an algorithm $\mathcal{A}$ with menu $\mathcal{M}(\mathcal{A})$ which achieves $\Omega(T)$ profile-swap regret in the worst case. Then, by definition, $\mathcal{M} \backslash \mathcal{M}_{NPSR} \neq \emptyset$. Let $\max_{x \in \mathcal{X}, y \in \mathcal{Y}} u_L(x, y) = u_L^{max}$. By Lemma 9, $\mathcal{M}_{NPSR}$ is the convex hull of all CSPs of the form $x \otimes y$, where $y \in \mathcal{Y}$ and $x \in \mathrm{BR}_L(y)$. Thus, $\mathcal{M}_{NPSR}^+$ is the set of all $x^* \otimes y^*$ pairs such that $u_L(x^*, y^*) = u_L^{max}$.

Now, let $\bar{\mathcal{M}} = \mathcal{M}(\mathcal{A}) \cup \mathcal{M}_{NPSR}^+$. Note that as $\mathcal{M}(\mathcal{A}) \subseteq \bar{\mathcal{M}}$, $\bar{\mathcal{M}} \setminus \mathcal{M}_{NPSR} \neq \emptyset$. Furthermore, by construction, $\mathcal{M}_{NPSR}^+ = \bar{\mathcal{M}}^+$. Thus, we can invoke Lemma 35 to show that there exists a u_O such that $V_L(\mathcal{M}_{NPSR}, u_O) > V_L(\bar{\mathcal{M}}, u_O)$. Now, there are two possibilities:

- The extreme point of u_O in $\bar{\mathcal{M}}$ is the same as that in $\mathcal{M}(\mathcal{A})$. Then $V_L(\bar{\mathcal{M}}, u_O) = V_L(\mathcal{M}(\mathcal{A}), u_O) < V_L(\mathcal{M}_{NPSR}, u_O)$.
- The extreme point of u_O in $\bar{\mathcal{M}}$ is *not* the same as that in $\mathcal{M}(\mathcal{A})$. Then, the extreme point of u_O in $\bar{\mathcal{M}}$ must be in $\bar{\mathcal{M}} \setminus \mathcal{M}(\mathcal{A})$, as otherwise it would also be selected in $\mathcal{M}(\mathcal{A})$. But for all points $(x, y) \in \bar{\mathcal{M}} \setminus \mathcal{M}(\mathcal{A})$, $u_L(x, y) = u_L^{max}$, and thus $V_L(\mathcal{M}_{NPSR}, u_O) > u_L^{max}$. This is a contradiction, and therefore this case cannot occur.

□

H.10 Proof of Lemma 21

Proof of Lemma 21. Note that the only way in which the optimizer's actions y_t are used in the definitions of external regret $\mathsf{Reg}(\mathbf{x}, \mathbf{y})$, linear swap regret $\mathsf{LinSwapReg}(\mathbf{x}, \mathbf{y})$, and polytope swap regret PolySwapReg$(\mathbf{x}, \mathbf{y})$ is to compute quantities of the form $u_L(\cdot, y_t)$. Since $r_t(x) = u_L(x, y_t)$, we can compute all such quantities given access to $r_t(x)$. □

H.11 Proof of Lemma 22

Proof of Lemma 22. Let $\mathbf{x} = (x_1, x_2, x_3, \dots, x_T) \in \mathcal{X}^T$ and $\mathbf{r} = (r_1, r_2, \dots, r_T) \in (\mathcal{X}^*)^T$. Consider the polytope game G where the action set $\mathcal{Y}$ of the optimizer is Δ_T (distributions over T pure actions), and the learner's utility function is given by $u_L(x, y) = \sum_{i=1}^{T} y_i r_i(x)$. Note that if we let $\mathbf{y} = (e_1, e_2, \dots, e_T)$ (where e_i is the ith unit vector), then $(\mathbf{x}, \mathbf{r})$ is the game-agnostic transcript corresponding to $(\mathbf{x}, \mathbf{y})$.

We now argue that ProfSwapReg$(\mathbf{x}, \mathbf{y})$ = PolySwapReg$(\mathbf{x}, \mathbf{y})$. The average CSP ϕ of the transcript $(\mathbf{x}, \mathbf{y})$ is given by

$$\phi = \frac{1}{T}\left(\sum_{t=1}^{T} x_t \otimes e_t\right). \tag{13}$$

Now, consider any decomposition of ϕ of the form

$$\phi = \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v (v \otimes y_v), \tag{14}$$

where $\lambda_v \geq 0$, $\sum_v \lambda_v = 1$, and $y_v \in \mathcal{Y}$. We claim that, for any such decomposition and any $t \in [T]$, we must have that $\sum_v \lambda_v y_{v,t} = 1/T$. To see this, consider the outcome when the (bi-affine) linear function $\rho : \mathcal{X} \otimes \mathcal{Y} \to \mathbb{R}$ defined via $\rho(x \otimes y) = y_t$ is evaluated on the (equal) right hand sides of both (13) and (14); for (13), this equals $1/T$, and for (14), this equals $\sum_v \lambda_v y_{v,t}$.

Given this, for each t, define $x_t^V \in \Delta(\mathcal{V}(\mathcal{X}))$ via $x_{t,v}^V = T\lambda_v y_{v,t}$. Note that since $\sum_v x_{t,v}^V = T\sum_v \lambda_v y_{v,t} = 1$, x_t^V does indeed belong to $\Delta(\mathcal{V}(\mathcal{X}))$. But now, from the definition of polytope swap regret, this specific action decomposition implies that

$$\begin{aligned}
\text{PolySwapReg}(\mathbf{x}, \mathbf{y}) &\leq \max_{\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \left(\sum_{t=1}^{T} u_L(\overline{\pi}(x_t^V), y_t) - u_L(x_t, y_t)\right) \\
&= T \cdot \max_{\pi : \mathcal{V}(\mathcal{X}) \to \mathcal{V}(\mathcal{X})} \left(\sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v u_L(\overline{\pi}(v), y_v) - u_L(v, y_v)\right).
\end{aligned}$$

Since ProfSwapReg$(\mathbf{x}, \mathbf{y})$ is equal to the minimum of this final expression over all possible decompositions of the form (14), this implies that PolySwapReg$(\mathbf{x}, \mathbf{y}) \leq$ ProfSwapReg$(\mathbf{x}, \mathbf{y})$. Together with the fact that ProfSwapReg$(\mathbf{x}, \mathbf{y}) \leq$ PolySwapReg$(\mathbf{x}, \mathbf{y})$ (Theorem 8), we have that PolySwapReg$(\mathbf{x}, \mathbf{y})$ = ProfSwapReg$(\mathbf{x}, \mathbf{y})$. □

H.12 Proof of Theorem 23

Proof of Theorem 23. For any polytope game G and transcript $(\mathbf{x}, \mathbf{y})$ in G, Theorem 8 tells us that ProfSwapReg$(\mathbf{x}, \mathbf{y}) \leq$ PolySwapReg$(\mathbf{x}, \mathbf{y})$. It follows that ProfSwapReg$_G(\mathcal{A}^T) \leq$ PolySwapReg$(\mathcal{A}^T)$ for any polytope game G (with the same learner action set $\mathcal{X}$ as $\mathcal{A}$).

Conversely, consider any game-agnostic transcript $(\mathbf{x}, \mathbf{r})$ of length T produced by $\mathcal{A}^T$. By Lemma 22, there exists a polytope game G and a (game-dependent) transcript $(\mathbf{x}, \mathbf{y})$ corresponding to $(\mathbf{x}, \mathbf{r})$ with the property that $\mathsf{ProfSwapReg}(\mathbf{x}, \mathbf{y}) = \mathsf{PolySwapReg}(\mathbf{x}, \mathbf{y})$. But now, note that if the optimizer plays the sequence of actions $\mathbf{y}$ against a learner employing the game-agnostic algorithm $\mathcal{A}$, the learner will see the sequence of reward functions $\mathbf{r}$ and therefore play the sequence of actions $\mathbf{x}$. It follows that there exists a polytope game G where $\mathsf{ProfSwapReg}_G(\mathcal{A}^T) \geq \mathsf{PolySwapReg}(\mathcal{A}^T)$. Together with the above result, this implies the theorem statement. □

H.13 Proof of Theorem 14

Proof of Theorem 14. Consider the vector-valued payoff function $v : \mathcal{X} \times \mathcal{Y} \to (\mathcal{X} \otimes \mathcal{Y}) \subset \mathbb{R}^{d_L d_O}$ defined via $v(x, y) = x \otimes y$, and let the target set $\mathcal{S}$ equal the no-profile-swap-regret menu $\mathcal{M}_{NPSR}$ for the game G. Note that v is response-satisfiable for $\mathcal{S}$, since given any $y \in \mathcal{Y}$, the CSPs in $\mathrm{BR}_L(y) \otimes y$ belong to $\mathcal{M}_{NPSR}$. Note also that $\|v\|_\infty \leq \max_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \|x \otimes y\| = (\max_{x \in \mathcal{X}} \|x\|)(\max_{y \in \mathcal{Y}} \|y\|) \leq 1$, since both $\mathcal{X}$ and $\mathcal{Y}$ are bounded in unit norm by assumption.

By Theorem 13, there exists a learning algorithm $\mathcal{A}$ which guarantees that $\mathrm{dist}\left(\frac{1}{T}\sum_{t=1}^{T} v(x_t, y_t), \mathcal{S}\right) \leq 2\|v\|_\infty \sqrt{T} \leq 2\sqrt{T}$. But $\frac{1}{T}\sum_{t=1}^{T} v(x_t, y_t) = \frac{1}{T}\sum_{t=1}^{T} x_t \otimes y_t$ is simply the CSP ϕ corresponding to this transcript of play; since $\mathcal{S} = \mathcal{M}_{NPSR}$, this distance is exactly the minimum distance from this CSP to the no-profile-swap-regret menu, and is therefore equal to $\mathsf{ProfSwapDist}(\phi)$. It follows that the worst-case profile swap distance $\mathsf{ProfSwapDist}(\mathcal{A}^T)$ is at most $2\sqrt{T}$. □

H.14 Proof of Theorem 16

Proof of Theorem 16. Assume to the contrary that such an efficient profile swap distance oracle exists. We first argue that we can use such an algorithm to construct a weak separation oracle for $\mathcal{M}_{NPSR}$ that also runs in time $\mathrm{poly}(n, m, c_L, c_O, 1/\delta)$.

This follows as a consequence of black-box convex oracle reductions established in [Lee et al., 2018]. In particular, our profile swap distance oracle is a weak evaluation oracle for the 1-Lipschitz, convex function $\mathsf{ProfSwapDist}(\phi)$. By Lemma 20 of [Lee et al., 2018], it is possible (with polynomially many calls to the original oracle) to extend this to a weak subgradient oracle for the same function. The weak subgradient oracle (when run with precision δ on input ϕ) returns a real number α satisfying $|\alpha - \mathsf{ProfSwapDist}(\phi)| \leq \delta$ and a vector $c \in \mathbb{R}^{d_L d_O}$ with the property that $\alpha + c^T(\phi' - \phi) \leq \mathsf{ProfSwapDist}(\phi') + \delta$ for all $\phi' \in \mathcal{X} \otimes \mathcal{Y}$.

Now, consider the halfspace given by the inequality $(\alpha - \delta) + c^T(\phi' - \phi) \leq 0$. We claim that this is a halfspace that δ-weakly separates $\mathcal{M}_{NPSR}$ from ϕ. In particular, every $\phi' \in \mathcal{M}_{NPSR}$ satisfies $\mathsf{ProfSwapDist}(\phi') = 0$, and thus satisfies this inequality. However, unless $\alpha - \delta \leq 0$, the CSP $\phi' = \phi$ does not satisfy this inequality. But if $\alpha - \delta \leq 0$, then since $|\alpha - \mathsf{ProfSwapDist}(\phi)| \leq \delta$, we must have that $\mathsf{ProfSwapDist}(\phi) \leq 2\delta$. We have therefore either provided a hyperplane separating $\mathcal{M}_{NPSR}$ from ϕ, or certified that ϕ is distance at most 2δ from $\mathcal{M}_{NPSR}$, completing the construction of our weak separation oracle.

With a weak separation oracle, we can construct a weak optimization oracle, and approximately optimize any linear function over $\mathcal{M}_{NPSR}$. But by Lemma 9, the optimal value of the linear function u_O over $\mathcal{M}_{NPSR}$ is $\mathsf{Stack}(G, u_O)$. Thus, this oracle would allow us to distinguish between the two cases in Lemma 15 in polynomial time, contradicting the APX-hardness of the result unless $P = NP$. □

H.15 Proof of Lemma 17

Proof of Lemma 17. For each unit vector $h \in \mathbb{R}^d$, consider the biaffine function $u_h(x, y) = \langle v(x, y), h \rangle - \sigma_h(\mathcal{S})$, where $\sigma_h(\mathcal{S}) = \max_{z \in \mathcal{S}} \langle h, z \rangle$. Let $\mathcal{U}$ be the convex hull over all such functions u_h.

If $\frac{1}{T}\sum_{t=1}^{T} u_h(x_t, y_t) \geq R$, then $\left\langle \frac{1}{T}\sum_{t=1}^{T} v(x_t, y_t), h \right\rangle - \sigma_h(\mathcal{S}) \geq R$, and therefore $\text{dist}\left(\frac{1}{T}\sum_{i=1}^{t} v(x_t, y_t), \mathcal{S}\right)$ is at least R. On the other hand, if $\text{dist}\left(\frac{1}{T}\sum_{i=1}^{t} v(x_t, y_t), \mathcal{S}\right)$ is at least R, then there must be a direction h in which $\left\langle \frac{1}{T}\sum_{t=1}^{T} v(x_t, y_t), h \right\rangle - \sigma_h(\mathcal{S}) \geq R$, and therefore we have $\frac{1}{T}\sum_{t=1}^{T} u_h(x_t, y_t) \geq R$. □

H.16 Proof of Lemma 19

Proof of Lemma 19. Consider an arbitrary bi-affine function $u_{\text{in}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that our semi-separation oracle receives as input. Compute via linear programming the minmax value $V = \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} u_{\text{in}}(x, y)$ of this game.

If $V \leq 0$, then this function is response-satisfiable (i.e., there exists an x^* such that $u_{\text{in}}(x^*, y) \leq 0$ for all $y \in \mathcal{Y}$), and we can return that u is response-satisfiable. Otherwise, there exists a $y^* \in \mathcal{Y}$ such that $u_{\text{in}}(x, y^*) \geq V > 0$ for all $x \in \mathcal{X}$, which we can again compute efficiently via a linear program.

Now, pick an arbitrary $x_{y^*} \in \text{BR}_L(y^*)$. Note that, by construction, $u(x_{y^*}, y^*) \leq 0$ for all $u \in \mathcal{U}$; on the other hand, for our queried u_{in}, we have that $u_{\text{in}}(x_{y^*}, y^*) \geq V > 0$. It follows that the linear constraint $u(x_{y^*}, y^*) \leq 0$ is a linear constraint on bilinear functions u separating u_{in} from the set $\mathcal{U}$. □

H.17 Proof of Theorem 20

For the purposes of discussing CE, we will depart from our learner/optimizer framework and will instead refer to the two players as Player 1 and Player 2, respectively.

Proof of Theorem 20. It suffices to check the conditions of Theorem 18 for the set $\mathcal{U}$ defined above. By assumption, our $\mathcal{X}$ and $\mathcal{Y}$ are bounded within the unit ball (so we can take $D_{\mathcal{X}} = D_{\mathcal{Y}} = 1$). The maximum norm of any element in our set $\mathcal{U}$ is at most $\sqrt{||h||^2 + |b|^2} \leq \sqrt{2}$, so we can take $D = \sqrt{2}$. Finally, note that any u with $||h|| \leq 1/2$ and $b \geq 1/2$ will satisfy $u(x, y) \leq 0$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$; we can therefore take $u_0(x, y) = -3/4$ (i.e., the u specified by $h = 0$ and $b = 3/4$) and $\rho = 1/4$. □

H.18 Proof of Theorem 24

We will need the following two lemmas.

Lemma 36. *Consider any* $v \in \mathcal{V}(\mathcal{X})$ *and* $\mathcal{V}(\mathcal{Y})_0 \subseteq \mathcal{V}(\mathcal{Y})$ *such that for all* $w_0 \in \mathcal{V}(\mathcal{Y})_0$, (v, w_0) *has support zero in any NFCE. Then, if* $(v, \hat{w}) \leq (\bar{v}, \hat{w})$ *for some* $\bar{v} \in \mathcal{V}(\mathcal{X})$ *and* $\forall \hat{w} \in \mathcal{V}(\mathcal{Y}) \setminus \mathcal{V}(\mathcal{Y})_0$, *then for all* $\hat{w} \in \mathcal{V}(\mathcal{Y})$ *such that* $(v, \hat{w}) < (\bar{v}, \hat{w})$, $(v, \hat{w})$ *has support zero in any NFCE.*

Proof. Consider some $(v, \hat{w})$ pair that satisfies the conditions above, and assume for contradiction that this pair has support $\alpha > 0$ in some NFCE ϕ^V. Conditioned on being recommended vertex v, Player 1's marginal distribution must be over only vertices in $\mathcal{V}(\mathcal{Y}) \setminus \mathcal{V}(\mathcal{Y})_0$, as otherwise a vertex pair (v, w_0) would have nonzero support on some NFCE. Therefore, the normal-form swap regret of this NFCE is

$$\begin{aligned}
\mathsf{NFSwapReg}_1(\phi^V) &= \max_{\pi^*:\mathcal{V}(\mathcal{X})\mapsto\mathcal{V}(\mathcal{X})} \sum_{\forall v_i}\sum_{\forall w_j} \phi^V_{v_i,w_j}\left(u_1(\phi^*(v_i),w_j)-u_1(v_i,w_j)\right)\\
&\geq \max_{\pi^*:\mathcal{V}(\mathcal{X})\mapsto\mathcal{V}(\mathcal{X})} \sum_{\forall w_j} \phi^V_{v,w_j}\left(u_1(\phi^*(v),w_j)-u_1(v,w_j)\right)\\
&\geq \sum_{\forall w_j} \phi^V_{v,w_j}\left(u_1(\bar{v},w_j)-u_1(v,w_j)\right)\\
&= \phi^V_{v,\hat{w}}\left(u_1(\bar{v},\hat{w})-u_1(v,\hat{w})\right) + \sum_{\forall w\neq\hat{w}} \phi^V_{v,w}\left(u_1(\bar{v},w)-u_1(v,w)\right)\\
&\geq \phi^V_{v,\hat{w}}\left(u_1(\bar{v},\hat{w})-u_1(v,\hat{w})\right)\\
&\qquad\text{(As by assumption, for all } w\in\mathcal{V}(\mathcal{Y})\text{, either } u_1(\bar{v},w)>u_1(v,w)\text{ or } \phi^V_{v,w}=0)\\
&= \alpha\left(u_1(\bar{v},\hat{w})-u_1(v,\hat{w})\right) > 0\\
&\qquad\text{(By our assumption on the support and utility of } (v,\hat{w}))
\end{aligned}$$

Thus, the normal-form-swap regret of our NFCE ϕ^V is > 0, leading to a contradiction. □

Lemma 37. *Consider any $w \in \mathcal{V}(\mathcal{Y})$ and $\mathcal{V}(\mathcal{X})_0 \subseteq \mathcal{V}(\mathcal{X})$ such that for all $v_0 \in \mathcal{V}(\mathcal{X})_0$, (v_0, w) has support zero in any NFCE. Then, if $(\hat{v}, w) \leq (\hat{v}, \bar{w})$ for some $\bar{w} \in \mathcal{V}(\mathcal{Y})$ and $\forall \hat{v} \in \mathcal{V}(\mathcal{X}) \setminus \mathcal{V}(\mathcal{X})_0$, then for all $\hat{v} \in \mathcal{V}(\mathcal{X})$ such that $(\hat{v}, w) < (\hat{v}, \bar{w})$, $(\hat{v}, w)$ has support zero in any NFCE.*

Proof. The proof is symmetric to the proof of Lemma 36. □

We can now prove Theorem 24.

Proof of Theorem 24. First, note that by Theorem 8, if $\phi = \mathrm{Proj}(\phi^V)$, we must have that $\mathsf{ProfSwapReg}_X(\phi) \leq \mathsf{NFSwapReg}_X(\phi^V)$ and $\mathsf{ProfSwapReg}_Y(\phi) \leq \mathsf{NFSwapReg}_Y(\phi^V)$. Therefore, if ϕ^V is a normal-form CE, it follows that ϕ is a profile CE.

We now construct a polytope game G where ϕ is a profile CE, but no normal-form CE ϕ^V has the property that $\mathrm{Proj}(\phi^V) = \phi$.

The game is as follows. The two players share the same action set $\mathcal{X} = \mathcal{Y} = [0,1]^2 \times \{1\}$, and so $\mathcal{V}(\mathcal{X}) = \mathcal{V}(\mathcal{Y}) = \{[0,0,1],[0,1,1],[1,1,1],[1,0,1]\}$. The utility function u_X of the $\mathcal{X}$-player is[13]

$$u_X = [-1,-1,0.5,0,1,0,-1,-1,-1].$$

The utility function of the $\mathcal{Y}$-player is

$$u_Y = [-0.5,1,1,-0.5,0.5,1,0.5,-1,-0.5]$$

Finally the CSP ϕ is

[13] Throughout this proof, we regularly flatten elements of $\mathcal{X} \otimes \mathcal{Y}$ to elements of $\mathbb{R}^9$ for convenience of notation. In particular, we use the notation that $v \otimes w = [v_1w_1, v_1w_2, v_1w_3, v_2w_1, \ldots, v_3w_3]$. We also represent bilinear functions by elements of $\mathbb{R}^9$ in a similar way; i.e., we define u_X and u_Y so that $u_X(\phi) = \langle u_x, \phi\rangle$.

$$\phi = [0, 0.2, 0.4, 0.2, 0.4, 0.6, 0.4, 0.4, 1].$$

The remainder of the proof is a computation that justifies the correctness of this counterexample.

While the utilities are a direct function of the underlying dimensions of $\mathcal{X}$ and $\mathcal{Y}$, we can also compute the utilities of the game defined by $\mathcal{V}(\mathcal{X})$ and $\mathcal{V}(\mathcal{Y})$ by computing, for each extreme point v and w in the vertex game, $u_X(\pi(v \otimes w))$. Doing so, we can write out the utilities of Player 1 and Player 2 in the vertex game:

$$\begin{array}{cccc} [0,0,1] & [0,1,1] & [1,1,1] & [1,0,1] \end{array}$$
$$\left(\begin{array}{cccc} -1 & -2 & -3 & -2 \\ -1 & -1 & -2 & -2 \\ -0.5 & -1.5 & -3.5 & -2.5 \\ -0.5 & -2.5 & -4.5 & -2.5 \end{array}\right) \begin{array}{c} [0,0,1] \\ [0,1,1] \\ [1,1,1] \\ [1,0,1] \end{array} \text{ Utility of Player 1}$$

$$\begin{array}{cccc} [0,0,1] & [0,1,1] & [1,1,1] & [1,0,1] \end{array}$$
$$\left(\begin{array}{cccc} -0.5 & -1.5 & -1 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 1.5 & 2 & 1.5 & 1 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{array}\right) \begin{array}{c} [0,0,1] \\ [0,1,1] \\ [1,1,1] \\ [1,0,1] \end{array} \text{ Utility of Player 2}$$

First, we must prove that ϕ is a PCE. To prove this, it is sufficient to show that:

- ϕ is a projection of a vertex CSP ϕ_1^V such that player 1 gets normal-form-swap regret of 0 under ϕ_1^V, and
- ϕ is a projection of a vertex CSP ϕ_2^V such that player 2 gets normal-form-swap regret of 0 under ϕ_2^V

As normal-form swap regret upper-bounds profile swap regret, this proves that both players achieve 0 PSR under ϕ, and thus that ϕ is a PCE.

For the first decomposition, consider the following vertex-game CSP ϕ_1^V:

$$\begin{array}{cccc} [0,0,1] & [0,1,1] & [1,1,1] & [1,0,1] \end{array}$$
$$\left(\begin{array}{cccc} 0.2 & 0 & 0 & 0.2 \\ 0 & 0 & 0.2 & 0 \\ 0.2 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \begin{array}{c} [0,0,1] \\ [0,1,1] \\ [1,1,1] \\ [1,0,1] \end{array}$$

It is easy to see that player 1 attains NFSR of 0 under ϕ_1^V by verifying that each of their actions in the vertex game is a best response against the conditional marginal distribution over player 2's actions against it. Furthermore,

$$\begin{aligned}
\mathrm{Proj}(\phi_1^V) = &\sum_{v\in\mathcal{V}(\mathcal{X}),w\in\mathcal{V}(\mathcal{Y})} \phi_1^V(v,w)(v\otimes w)\\
&= 0.2\cdot([0,0,1]\otimes[0,0,1]) + 0.2\cdot([0,0,1]\otimes[1,0,1]) + 0.2\cdot([0,1,1]\otimes[1,1,1])\\
&+ 0.2\cdot([1,1,1]\otimes[0,0,1]) + 0.2\cdot([1,1,1]\otimes[0,1,1])\\
&= 0.2\cdot([0,0,0,0,0,0,0,0,1]) + 0.2\cdot([0,0,0,0,0,0,1,0,1]) + 0.2\cdot([0,0,0,1,1,1,1,1,1])\\
&+ 0.2\cdot([0,0,1,0,0,1,0,0,1]) + 0.2\cdot([0,1,1,0,1,1,0,1,1])\\
=&[0,0.2,0.4,0.2,0.4,0.6,0.4,0.4,1] = \phi
\end{aligned}$$

For the second decomposition, consider the following vertex-game CSP ϕ_2^V:

$$\begin{array}{ccccc}
[0,0,1] & [0,1,1] & [1,1,1] & [1,0,1] & \\
0 & 0 & 0 & 0.2 & [0,0,1]\\
0 & 0.2 & 0 & 0.2 & [0,1,1]\\
0 & 0.2 & 0 & 0 & [1,1,1]\\
0.2 & 0 & 0 & 0 & [1,0,1]
\end{array}$$

It is again easy to see that player 2 attains NFSR of 0 under ϕ_2^V by verifying that each of their actions in the vertex game is a best response against the conditional marginal distribution over player 1's actions against it. Furthermore, we can verify that it is indeed a decomposition of ϕ by again computing its projection:

$$\begin{aligned}
\mathrm{Proj}(\phi_2^V) = &\sum_{v\in\mathcal{V}(\mathcal{X}),w\in\mathcal{V}(\mathcal{Y})} \phi_2^V(v,w)(v\otimes w)\\
&= 0.2\cdot([0,0,1]\otimes[1,0,1]) + 0.2\cdot([0,1,1]\otimes[0,1,1]) + 0.2\cdot([0,1,1]\otimes[1,0,1])\\
&+ 0.2\cdot([1,1,1]\otimes[0,1,1]) + 0.2\cdot([1,0,1]\otimes[0,0,1])\\
&= 0.2\cdot([0,0,0,0,0,0,1,0,1]) + 0.2\cdot([0,0,0,0,1,1,0,1,1]) + 0.2\cdot([0,0,0,1,0,1,1,0,1])\\
&+ 0.2\cdot([0,1,1,0,1,1,0,1,1]) + 0.2\cdot([0,0,1,0,0,0,0,0,1])\\
=&[0,0.2,0.4,0.2,0.4,0.6,0.4,0.4,1] = \phi
\end{aligned}$$

Therefore, $\phi = [0,0.2,0.4,0.2,0.4,0.6,0.4,0.4,1]$ is a PCE.

Next, we will show that there does not exist a NFCE σ such that $\pi(\sigma) = \phi$. To prove this, we assume for contradiction that there does exist such an NFCE, and iteratively eliminate move pairs from its possible support until we find a contradiction. To do this, we will crucially make use of Lemmas 36 and 37, which formalize the fact that not only can't a strictly dominated action be present in an NFCE, if there is a weakly dominated action[14], the action pairs where it is strictly worse than the dominating action also cannot be in any NFCE. These properties hold even if the domination is only over the support of the NFCE itself.

It will be clarifying to keep track of this visually: we begin by allowing the possibility of all possible move pairs to be in the support of some NFCE, denoting this by a ? denoting the weight placed

[14] Here we use *strictly* and *weakly dominated* in a slightly different way than in when excluding weakly dominated actions from the game; here we say action a strictly dominates b when another a is strictly better than b for all actions of the opponent, and a weakly dominates b when it is weakly better for all actions, and strictly better for at least one.

upon each move pair. As we prove that certain vertex action pairs cannot be in the support of any NFCE, we will replace these ?s with 0s. For simplicity, we will also re-label the vertices as $v_{1:4}$, $w_{1:4}$. We begin by noting that since the first index of ϕ is 0, the following entries are all zero:

$$\begin{array}{cccccc} & w_1 & w_2 & w_3 & w_4 & \\ \left(\right. & ? & ? & ? & ? & \left.\right) v_1 \\ & ? & ? & ? & ? & v_2 \\ & ? & ? & \mathbf{0} & \mathbf{0} & v_3 \\ & ? & ? & \mathbf{0} & \mathbf{0} & v_4 \end{array}$$

First, note that v_1 is dominated for player 1 by v_2, and strictly dominated in two of the columns. Thus, by Lemma 36, (v_1, w_2) and (v_1, w_3) must have support zero in any NFCE.

Furthermore, v_4 is dominated for Player 1 by v_3, and strictly dominated in two of the columns. Therefore, again by Lemma 36, (v_4, w_2) and (v_4, w_3) must have support zero in any NFCE. Additionally, conditional on player 1 being recommended v_4 in a NFCE satisfying our constraints, player 2 must have support only on w_1 and w_2. Therefore by Lemma 37, (v_4, w_2) cannot be in the support either.

$$\begin{array}{c} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \\ \left(\begin{array}{cccc} ? & \mathbf{0} & \mathbf{0} & ? \\ ? & ? & ? & ? \\ ? & ? & 0 & 0 \\ ? & \mathbf{0} & 0 & 0 \end{array}\right) \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

Now, note that conditional on player 2 being recommended w_3, player 1 must have full support on v_2. Player 2 would rather play w_1 in this case, so by Lemma 37 we can eliminate another vertex pair from the support.

$$\begin{array}{c} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \\ \left(\begin{array}{cccc} ? & 0 & 0 & ? \\ ? & ? & \mathbf{0} & ? \\ ? & ? & 0 & 0 \\ ? & 0 & 0 & 0 \end{array}\right) \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

From here, we can label the remaining vertex pairs in order to determine additional constraints:

$$\begin{array}{c} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \\ \left(\begin{array}{cccc} \mathrm{a} & 0 & 0 & \mathrm{b} \\ \mathrm{c} & \mathrm{d} & 0 & \mathrm{e} \\ \mathrm{f} & \mathrm{g} & 0 & 0 \\ \mathrm{h} & 0 & 0 & 0 \end{array}\right) \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

Recall that $\phi[4] = 0.2$, and further that $\phi[4]$ is the sum of the values at the intersection of the leftmost two columns and the middle rwo rows. Therefore, this tells us that $e = 0.2$. Furthermore, $\phi[2] = 0.2$ as well, and this value represents the sum of the four values in the intersection of the middle two columns and the lowest two rows. Thus, $g = 0.2$. Also, $\phi[5] = 0.4$, representing the central four values in the matrix. Thus, $d + g = 0.4$, and therefore $d = 0.2$.

$$\begin{array}{cc} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} & \\ \begin{pmatrix} a & 0 & 0 & b \\ c & \mathbf{0.2} & 0 & \mathbf{0.2} \\ f & \mathbf{0.2} & 0 & 0 \\ h & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

Also, $\phi[6] = 0.6$, which represents the middle two rows of the matrix. Therefore, c and f must both equal 0:

$$\begin{array}{cc} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} & \\ \begin{pmatrix} a & 0 & 0 & b \\ \mathbf{0} & 0.2 & 0 & 0.2 \\ \mathbf{0} & 0.2 & 0 & 0 \\ h & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

And $\phi[3] = 0.4$, which represents the bottom two rows. There is only one variable remaining in these rows we can set, and thus $h = 0.2$:

$$\begin{array}{cc} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} & \\ \begin{pmatrix} a & 0 & 0 & b \\ 0 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 0 & 0 \\ \mathbf{0.2} & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

Next, $\phi[7] = 0.4$, which represents the leftmost two columns. Thus, $b = 0.2$:

$$\begin{array}{cc} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} & \\ \begin{pmatrix} a & 0 & 0 & \mathbf{0.2} \\ 0 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 0 & 0 \\ 0.2 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

Finally, given that these values must all sum to 1, this leaves us with a unique solution:

$$\begin{array}{cc} \begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} & \\ \begin{pmatrix} \mathbf{0} & 0 & 0 & 0.2 \\ 0 & 0.2 & 0 & 0.2 \\ 0 & 0.2 & 0 & 0 \\ 0.2 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \end{array}$$

However, this is *not* an NFCE. Conditioned on player 1 playing v_3, player 1's marginal distribution is full support on w_2. But in this case, player 1 has normal-form-swap regret to playing v_2. This leads to a contradiction, and therefore ϕ is not the projection of any NFCE.

Furthermore, we verified via a linear program that there does not exist any other ϕ^* which is the projection of some NFCE, and has the same utility profile as ϕ. □

H.19 Proof of Theorem 25

Proof of Theorem 25. Since $\mathsf{ProfSwapReg}_X(\phi) = 0$, we can write

$$\phi = \sum_{v \in \mathcal{V}(\mathcal{X})} \lambda_v (v \otimes y_v). \tag{15}$$

for some choice of $y_v \in \mathcal{Y}$ and nonnegative λ_v summing to one. Similarly, since $\mathsf{ProfSwapReg}_Y(\phi) = 0$, we can write

$$\phi = \sum_{w \in \mathcal{V}(\mathcal{Y})} \mu_w (x_w \otimes w). \tag{16}$$

We claim that, when $\mathcal{X}$ is a simplex, it is possible to decompose each x_w into a combination of elements in $\mathcal{V}(\mathcal{X})$ and each y_v into a combination of elements in $\mathcal{V}(\mathcal{Y})$ in such a way that these two decompositions agree. That is, we can write $x_w = \sum_{v \in \mathcal{V}(\mathcal{X})} \gamma_{v,w} v$ and $y_v = \sum_{w \in \mathcal{V}(\mathcal{Y})} \gamma'_{v,w} w$ so that, for each $(v, w) \in \mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y})$,

$$\lambda_v \gamma'_{v,w} = \mu_w \gamma_{v,w}, \tag{17}$$

and we can therefore write

$$\phi = \sum_{v \in \mathcal{V}(\mathcal{X})} \sum_{w \in \mathcal{V}(\mathcal{Y})} \zeta_{v,w} (v \otimes w)$$

where $\zeta_{v,w}$ is the common value of the two sides of (17). But now, note that this gives rise to a normal-form CE $\phi^V \in \Delta(\mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{Y}))$ via $\phi^V(v, w) = \zeta_{v,w}$. In particular, by (15), we have that $\mathsf{NFSwapReg}_X(\phi^V) = 0$, and by (16), we have that $\mathsf{NFSwapReg}_Y(\phi^V) = 0$.

It remains to show that a common decomposition (i.e., of the form in (17)) exists. To see this, note that since $\mathcal{X}$ is a simplex, there is a unique way to write each x_w as a convex combination $\sum_{v \in \mathcal{V}(\mathcal{X})} \gamma_{v,w} v$. We claim that if we then define $\gamma'_{v,w} = (\mu_w \gamma_{v,w})/\lambda_v$ (so to satisfy (17)), it must be the case that $y_v = \sum_{w \in \mathcal{V}(\mathcal{Y})} \gamma'_{v,w} w$.

To see this, for any $v \in \mathcal{V}(\mathcal{X})$, consider the bilinear function $\rho_v : \mathcal{X} \otimes \mathcal{Y} \to \mathbb{R}\mathcal{Y}$ defined via $\rho_v(v \otimes w) = w$ and $\rho_v(v' \otimes w) = 0$ for $v' \neq w$ (note that this function only exists since $\mathcal{X}$ is a simplex). Applying ρ_v to (15), we have that $\rho_v(\phi) = \lambda_v y_v$. On the other hand, applying ρ_v to (16) (after substituting in our above decomposition for x), we have that $\rho_v(\phi) = \sum_w \mu_w \gamma_{v,w} w$. Equating these two expressions for $\rho_v(\phi)$, it follows that $y_v = \sum_w \gamma'_{v,w} w$, as desired. □