Pareto-Optimal Algorithms for Learning in Games

ESHWAR RAM ARUNACHALESWARAN, University of Pennsylvania, USA
NATALIE COLLINA, University of Pennsylvania, USA
JON SCHNEIDER, Google Research, USA

We study the problem of characterizing optimal learning algorithms for playing repeated games against an adversary with unknown payoffs. In this problem, the first player (called the learner) commits to a learning algorithm against a second player (called the optimizer), and the optimizer best-responds by choosing the optimal dynamic strategy for their (unknown but well-defined) payoff. Classic learning algorithms (such as no-regret algorithms) provide some counterfactual guarantees for the learner, but might perform much more poorly than other learning algorithms against particular optimizer payoffs.

In this paper, we introduce the notion of *asymptotically Pareto-optimal learning algorithms*. Intuitively, if a learning algorithm is Pareto-optimal, then there is no other algorithm which performs asymptotically at least as well against all optimizers and performs strictly better (by at least $\Omega(T)$) against some optimizer. We show that well-known no-regret algorithms such as Multiplicative Weights and Follow The Regularized Leader are Pareto-dominated. However, while no-regret is not enough to ensure Pareto-optimality, we show that a strictly stronger property, no-swap-regret, is a sufficient condition for Pareto-optimality.

Proving these results requires us to address various technical challenges specific to repeated play, including the fact that there is no simple characterization of how optimizers who are rational in the long-term best-respond against a learning algorithm over multiple rounds of play. To address this, we introduce the idea of the *asymptotic menu* of a learning algorithm: the convex closure of all correlated distributions over strategy profiles that are asymptotically implementable by an adversary. Interestingly, we show that all no-swap-regret algorithms share the same asymptotic menu, implying that all no-swap-regret algorithms are "strategically equivalent".

CCS Concepts: • **Theory of computation → Theory and algorithms for application domains**; **Algorithmic game theory and mechanism design**;

Additional Key Words and Phrases: Learning in Games, Non-Myopic Best-Responses, Algorithms as Strategies, Stackelberg Equilibria

ACM Reference Format:
Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. 2024. Pareto-Optimal Algorithms for Learning in Games. In *Conference on Economics and Computation (EC '24), July 8–11, 2024, New Haven, CT, USA.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3670865.3673517

1 Introduction

Consider an agent faced with the problem of playing a repeated game against another strategic agent. In the absence of complete information about the other agent's goals and behavior, it is reasonable for the agent to employ a learning algorithm to decide how to play. This raises the (purposefully vague) question: What is the "best" learning algorithm for learning in games?

Authors' Contact Information: Eshwar Ram Arunachaleswaran, eshwarram.arunachaleswaran@gmail.com, University of Pennsylvania, Philadelphia, USA; Natalie Collina, ncollina@seas.upenn.edu, University of Pennsylvania, Philadelphia, USA; Jon Schneider, jschnei@google.com, Google Research, New York, USA.

EC '24, July 8–11, 2024, New Haven, CT, USA

ACM ISBN 979-8-4007-0704-9/24/07
https://doi.org/10.1145/3670865.3673517

One popular yardstick for measuring the quality of learning algorithms is *regret*. The regret of a learning algorithm is the worst-case gap between the algorithm's achieved utility and the counterfactual utility it would have received if it instead had played the optimal fixed strategy in hindsight. There exist learning algorithms which achieve sublinear $o(T)$ regret when played across T rounds (*no-regret algorithms*), and researchers now have a very good understanding of the strongest regret guarantees possible in a variety of different settings. It is tempting to conclude that one of these regret-minimizing algorithms is the optimal choice of learning algorithm for our agent.

However, many standard no-regret algorithms – including popular algorithms such as Multiplicative Weights and Follow-The-Regularized-Leader – have the unfortunate property that they are vulnerable to strategic manipulation [Deng et al., 2019b]. What this means is that if one agent (a learner) is using a such an algorithm to play a repeated game against a second agent (an optimizer), there are games where the optimizer can exploit this by playing a time-varying dynamic strategy (e.g. playing some strategy for the first $T/2$ rounds, then switching to a different strategy in the last half of the rounds). By doing so, in some games the optimizer can obtain significantly more ($\Omega(T)$) utility than they could by playing a fixed static strategy, often at cost to the learner. This is perhaps most striking in the case of auctions, where [Braverman et al., 2018] show that if a bidder uses such an algorithm to decide their bids, the auctioneer can design a dynamic auction that extracts the full welfare of the bidder as revenue (leaving the bidder with zero net utility). On the flip side, [Deng et al., 2019b] show that if the learner employs a learning algorithm with a stronger counterfactual guarantee – that of *no-swap-regret* – this protects the learner from strategic manipulation, and prevents the optimizer from doing anything significantly better than playing a static strategy for all T rounds. Perhaps, then, a no-swap-regret algorithm is the "best" learning algorithm for game-theoretic settings.

But even this is not the complete picture: even though strategic manipulation from the other agent *may* harm the learner, there are other games where both the learner and optimizer can benefit from the learner playing a manipulable algorithm. Indeed, [Guruganesh et al., 2024] prove that there are contract-theoretic settings where both the learner and optimizer benefit from from the learner running a manipulable no-regret algorithm (with the optimizer best-responding to it). In light of these seemingly contradictory results, is there anything meaningful one can say about what learning algorithm a strategic agent should use?

1.1 Our results and techniques

In this paper, we acknowledge that there may not be a consistent total ordering among learning algorithms, and instead study this question through the lens of *Pareto-optimality*. Specifically, we consider the following setting. As before, one agent (the learner) is repeatedly playing a general-sum normal-form game G against a second agent (the optimizer). The learner knows their own utility u_L for outcomes of this game but is uncertain of the optimizer's utility u_O, and so commits to playing according to a learning algorithm $\mathcal{A}$ (a procedure which decides the learner's action at round t as a function of the observed history of play of both parties). The optimizer observes this and plays a (potentially dynamic) best-response to $\mathcal{A}$ that maximizes their own utility u_O. We remark that in addition to capturing the strategic settings mentioned above, this asymmetry also models settings where one of the participants in a repeated game (such as a market designer or large corporation) must publish their algorithms up front and has to play against a large collection of unknown optimizers.

We say one learning algorithm $\mathcal{A}$ *(asymptotically) Pareto-dominates* a second learning algorithm $\mathcal{A}'$ for the learner in this game if: i. for any utility function the optimizer may have, the learner receives at least as much utility (up to sublinear $o(T)$ factors) under committing to $\mathcal{A}$ as they do

under committing to $\mathcal{A}'$, and ii. there exists at least one utility function for the optimizer where the learner receives significantly more utility (at least $\Omega(T)$ more) by committing to $\mathcal{A}$ instead of committing to $\mathcal{A}'$. A learning algorithm $\mathcal{A}$ which is not Pareto-dominated by any learning algorithm is *Pareto-optimal*.

We prove the following results about Pareto-domination of learning algorithms for games.

- First, our notion of Pareto-optimality is non-vacuous: there exist many learning algorithms (including many no-regret learning algorithms) which are Pareto-dominated. In fact, we can show that there exist large classes of games where any instantiation of the Follow-The-Regularized-Leader (FTRL) with a strongly convex regularizer is Pareto-dominated. This set of learning algorithms contains many of the most popular no-regret learning algorithms as special cases (e.g. Hedge, Multiplicative Weights, and Follow-The-Perturbed-Leader).
- In contrast to this, any no-swap-regret algorithm is Pareto-optimal. This strengthens the case for the strategic power of no-swap-regret learning algorithms in repeated interactions.
- That said, the Pareto-domination hierarchy of algorithms is indeed not a total order: there exist infinitely many Pareto-optimal learning algorithms that are qualitatively different in significant ways. And of the learning algorithms that are Pareto-dominated, they are not all Pareto-dominated by the same Pareto-optimal algorithm (indeed, in many cases FTRL is not dominated by a no-swap-regret learning algorithm, but by a different Pareto-optimal learning algorithm).

In addition to this, we also provide a partial characterization of all no-regret Pareto-optimal algorithms, which we employ to prove the above results. In order to understand this characterization, we need to introduce the notion of the *asymptotic menu* of a learning algorithm.

To motivate this concept, consider a transcript of the repeated game G. If the learner has m actions to choose from each round and the optimizer has n actions, then after playing for T rounds, we can describe the average outcome of play via a *correlated strategy profile (CSP)*: a correlated distribution over the mn pairs of learner/optimizer actions. The important observation is that this correlated strategy profile (an mn-dimensional object) is all that is necessary to understand the average utilities of both players, regardless of their specific payoff functions – it is in some sense a "sufficient statistic" for all utility-theoretic properties of the transcript.

Inspired by this, we define the *asymptotic menu* $\mathcal{M}(\mathcal{A})$ of a learning algorithm $\mathcal{A}$ to be the convex closure of the set of all CSPs that are asymptotically implementable by an optimizer against a learner who is running algorithm $\mathcal{A}$. That is, a specific correlated strategy profile φ belongs to $\mathcal{M}(\mathcal{A})$ if the optimizer can construct arbitrarily long transcripts by playing against $\mathcal{A}$ whose associated CSPs are arbitrarily close to φ. We call this a "menu" since we can think of this as a set of choices the learner offers the optimizer by committing to algorithm $\mathcal{A}$ (essentially saying, "pick whichever CSP in this set you prefer the most").

Working with asymptotic menus allows us to translate statements about learning algorithms (complex, ill-behaved objects) to statements about convex subsets of the mn-simplex (much nicer mathematical objects). In particular, our notion of Pareto-dominance translates directly from algorithms to menus, as do concepts like "no-regret" and "no-swap-regret". This allows us to prove the following results about asymptotic menus:

- First, by applying Blackwell's Approachability Theorem, we give a simple and complete characterization of which convex subsets of the mn-simplex Δ_{mn} are valid asymptotic menus: any set $\mathcal{M}$ with the property that for any optimizer action $y \in \Delta_n$, there exists a learner action $x \in \Delta_m$ such that the product distribution $x \otimes y$ belongs to $\mathcal{M}$ (Theorem 3.3).
- We then use this characterization to show that there is a *unique* no-swap-regret menu, which we call $\mathcal{M}_{NSR}$ (Theorem 3.9), can be described explicitly as a polytope, and which is contained

as a subset of *any* no-regret menu (Lemma 3.8). In particular, this implies that all no-swap-regret algorithms share the same asymptotic menu, and hence are *strategically equivalent* from the perspective of an optimizer strategizing against them. This is notably not the case for no-regret algorithms, which have many different asymptotic menus.

- In our main result, we give a characterization of all Pareto-optimal no-regret menus for menus that are polytopes (the intersection of finitely many half-spaces). We show that such a menu $\mathcal{M}$ is Pareto-optimal iff the set of points in $\mathcal{M}$ which minimize the learner's utility are the same as that for the no-swap-regret menu $\mathcal{M}_{NSR}$ (Theorem 4.1). It is here where our geometric view of menus is particularly useful: it allows us to reduce this general question to a non-trivial property of two-dimensional convex curves (Lemma 4.5).
- As an immediate consequence of this, we show the no-swap-regret menu (and hence any no-swap-regret algorithm) is Pareto-optimal (Corollary A.1), and that there exist infinitely many distinct Pareto-optimal menus (each of which can be formed by starting with the no-swap-regret menu $\mathcal{M}_{NSR}$ and expanding it to include additional no-regret CSPs).
- Finally, we demonstrate instances where the asymptotic menu of FTRL is Pareto-dominated. This would follow nearly immediately from the characterization above (in fact, for the even larger class of *mean-based* no-regret algorithms), but for the restriction that the above characterization only applies to polytopal menus. To handle this, we also find a class of examples where we can prove that the asymptotic menu of FTRL is a polytope. Doing this involves developing new tools for optimizing against mean-based algorithms, and may be of independent interest.

1.2 Takeaways and future directions

What do these results imply about our original question? Can we say anything new about which learning algorithms a learner should use to play repeated games? From a very pessimistic point of view, the wealth of Pareto-optimal algorithms means that we cannot confidently say that any specific algorithm is the "best" algorithm for learning in games. But more optimistically, our results clearly highlight no-swap-regret learning algorithms as a particularly fundamental class of learning algorithms in strategic settings (in a way generic no-regret algorithms are not), with the no-swap-regret menu being the *minimal* Pareto-optimal menu among all no-regret Pareto-optimal menus.

We would also argue that our results do have concrete implications for how one should think about designing new learning algorithms in game-theoretic settings. In particular, they suggest that instead of directly designing a learning algorithm via minimizing specific regret notions (which can lead to learning algorithms which are Pareto-dominated, in the case of common no-regret algorithms), it may be more fruitful to first design the specific asymptotic menu we wish the algorithm to converge to (and only then worry about the rate at which algorithms approach this menu). Our characterization of Pareto-optimal no-regret menus provides a framework under which we can do this: start with the no-swap-regret menu, and expand it to contain other CSPs that we believe may be helpful for the learner. For example, consider a learner who believes that the optimizer has a specific utility function u_O, but still wants to run a no-regret learning algorithm to hedge against the possibility that they do not. This learner can use our characterization to first find the best such asymptotic menu, and then construct an efficient learning algorithm that approaches it (via e.g. the Blackwell approachability technique of Theorem 3.3).

There are a number of interesting future directions to explore. Most obvious is the question of extending our characterization of Pareto-optimality from polytopal no-regret menus to all asymptotic menus. While we conjecture the polytopal constraint is unnecessary, there do exist

non-trivial high-regret Pareto-optimal menus (Theorem 4.9), and understanding the full class of such menus is an interesting open problem.

Secondly, throughout this discussion we have taken the perspective of a learner who is aware of their own payoff u_L and only has uncertainty about the optimizer they face. Yet one feature of most common learning algorithms is that they do not even require this knowledge about u_L, and are designed to work in a setting where they learn their own utilities over time. Some of our results (such as the Pareto-optimality of no-swap-regret algorithms) carry over near identically to such utility-agnostic settings (see the full version [Arunachaleswaran et al., 2024]), but we still lack a clear understanding of Pareto-domination there.

Finally, we focus entirely on normal-form two-player games. But many practical applications of learning algorithms take place in more general strategic settings, such as Bayesian games or extensive-form games. What is the correct analogue of asymptotic menus and Pareto-optimality for these settings?

1.3 Related work

There is a long history of work in both economics and computer science of understanding the interplay between game theory and learning. We refer the reader to any of [Cesa-Bianchi and Lugosi, 2006, Fudenberg and Levine, 1998, Young, 2004] for an introduction to the area. Much of the recent work in this area is focused on understanding the correlated equilibria that arise when several learning algorithms play against each other, and designing algorithms which approach this set of equilibria more quickly or more stably (e.g., [Anagnostides et al., 2022a,b, Farina et al., 2022, Piliouras et al., 2022, Syrgkanis et al., 2015, Zhang et al., 2023]). It may be helpful to compare the learning-theoretic characterization of the set of correlated equilibria (which contains all CSPs that can be implemented by having several no-swap-regret algorithms play against each other) to our definition of asymptotic menu – in some ways, one can think of an asymptotic menu as a one-sided, algorithm-specific variant of this idea.

Our paper is most closely connected to a growing area of work on understanding the strategic manipulability of learning algorithms in games. [Braverman et al., 2018] was one of the first works to investigate these questions, specifically for the setting of non-truthful auctions with a single buyer. Since then, similar phenomena have been studied in a variety of economic settings, including other auction settings [Cai et al., 2023, Deng et al., 2019a, Kolumbus and Nisan, 2022a,b], contract design [Guruganesh et al., 2024], Bayesian persuasion [Chen and Lin, 2023], general games [Brown et al., 2023, Deng et al., 2019b], and Bayesian games [Mansour et al., 2022]. [Deng et al., 2019b] and [Mansour et al., 2022] show that no-swap-regret is a necessary and sufficient condition to prevent the optimizer from benefiting by manipulating the learner. [Brown et al., 2023] introduce an asymmetric generalization of correlated and coarse-correlated equilibria which they use to understand when learners are incentivized to commit to playing certain classes of learning algorithms. Our no-regret and no-swap-regret menus can be interpreted as the sets of $(\emptyset, \mathcal{E})$-equilibria and $(\emptyset, \mathcal{I})$-equilibria in their model (their definition of equilibria stops short of being able to express the asymptotic menu of a specific learning algorithm, however). In constructing an example where the asymptotic menu of FTRL is a polytope, we borrow an example from [Guruganesh et al., 2024], who present families of principal-agent problems which are particularly nice to analyze from the perspective of manipulating mean-based agents.

Our results highlight no-swap-regret algorithms as particularly relevant algorithms for learning in games. The first no-swap-regret algorithms were provided by [Foster and Vohra, 1997], who also showed their dynamics converge to correlated equilibria. Since then, several authors have designed learning algorithms for minimizing swap regret in games [Blum and Mansour, 2007, Dagan et al., 2023, Hart and Mas-Colell, 2000, Peng and Rubinstein, 2023]. Our work shows that

all these algorithms are in some "strategically equivalent" up to sublinear factors; this is perhaps surprising given that many of these algorithms are qualitatively quite different (especially the very recent swap regret algorithms of [Peng and Rubinstein, 2023] and [Dagan et al., 2023]).

Finally, although we phrase our results from the perspective of learning in games, it is equally valid to think of this work as studying a Stackelberg variant of a repeated, finite-horizon game, where one player must commit to a repeated strategy without being fully aware of the other player's utility function. In the full-information setting (where the learner is aware of the optimizer's payoff), the computational aspects of this problem are well-understood [Collina et al., 2023, Conitzer and Sandholm, 2006, Peng et al., 2019]. In the unknown-payoff setting, preexisting work has focused on learning the optimal single-round Stackelberg distribution by playing repeatedly against a myopic [Balcan et al., 2015, Lauffer et al., 2022, Marecki et al., 2012] or discounting [Haghtalab et al., 2022] follower. As far as we are aware, we are the first to study this problem in the unknown-payoff setting with a fully non-myopic follower.

2 Model and preliminaries

We consider a setting where two players, an *optimizer* O and a *learner* L, repeatedly play a two-player bimatrix game G for T rounds. The game G has m actions for the optimizer and n actions for the learner, and is specified by two bounded payoff functions $u_O : [m] \times [n] \to [-1, 1]$ (denoting the payoff for the optimizer) and $u_L : [m] \times [n] \to [-1, 1]$ (denoting the payoff for the learner). During each round t, the optimizer picks a mixed strategy $x_t \in \Delta_m$ while the learner simultaneously picks a mixed strategy $y_t \in \Delta_n$; the learner then receives reward $u_L(x_t, y_t)$ and the optimizer receives reward $u_O(x_t, y_t)$ (where here we have linearly extended u_L and u_O to take domain $\Delta_m \times \Delta_n$). Both the learner and optimizer observe the full mixed strategy of the other player (the "full-information" setting).

True to their name, the learner will employ a *learning algorithm* $\mathcal{A}$ to decide how to play. For our purposes, a learning algorithm is a family of horizon-dependent algorithms $\{A^T\}_{T \in \mathbb{N}}$. Each A^T describes the algorithm the learner follows for a fixed time horizon T. Each horizon-dependent algorithm is a mapping from the history of play to the next round's action, denoted by a collection of T functions $A_1^T, A_2^T \cdots A_T^T$, each of which deterministically map the transcript of play (up to the corresponding round) to a mixed strategy to be used in the next round, i.e., $A_t^T(x_1, x_2, \cdots, x_{t-1}) = y_t$.

We assume that the learner is able to see u_L before committing to their algorithm $\mathcal{A}$, but not u_O. The optimizer, who knows u_L and u_O, will approximately best-respond by selecting a sequence of actions that approximately (up to sublinear $o(T)$ factors) maximizes their payoff. They break ties in the learner's favor. Formally, for each T let

$$V_O(\mathcal{A}, u_O, T) = \sup_{(x_1,\ldots,x_T) \in \Delta_m^T} \frac{1}{T} \sum_{t=1}^{T} u_O(x_t, y_t)$$

represent the maximum per-round utility of the optimizer with payoff u_O playing against $\mathcal{A}$ in a T round game (here and throughout, each y_t is determined by running A_t^T on the prefix x_1 through x_{t-1}). For any $\varepsilon > 0$, let

$$\mathcal{X}(\mathcal{A}, u_O, T, \varepsilon) = \left\{ (x_1, x_2, \ldots, x_T) \in \Delta_m^T \;\middle|\; \frac{1}{T} \sum_{t=1}^{T} u_O(x_t, y_t) \geq V_O(\mathcal{A}, u_O, T) - \varepsilon \right\}$$

be the set of ε-approximate best-responses for the optimizer to the algorithm $\mathcal{A}$. Finally, let

$$V_L(\mathcal{A}, u_O, T, \varepsilon) = \sup_{(x_1,\ldots,x_T)\in\mathcal{X}(\mathcal{A},u_O,T,\varepsilon)} \frac{1}{T}\sum_{t=1}^{T} u_L(x_t, y_t)$$

represent the maximum per-round utility of the learner under any of these approximate best-responses.

We are concerned with the asymptotic per-round payoff of the learner as $T \to \infty$ and $\varepsilon \to 0$. Specifically, let

$$V_L(\mathcal{A}, u_O) = \lim_{\varepsilon\to 0} \liminf_{T\to\infty} V_L(\mathcal{A}, u_O, T, \varepsilon). \tag{1}$$

Note that the outer limit in (1) is well-defined since for each T, $V_L(A, u_O, T, \varepsilon)$ is decreasing in ε (being a supremum over a smaller set).

The learner would like to select a learning algorithm $\mathcal{A}$ that is "good" regardless of what the optimizer payoffs u_O are. In particular, the learner would like to choose a learning algorithm that is *asymptotically Pareto-optimal* in the following sense.

Definition 2.1 (Asymptotic Pareto-dominance for learning algorithms). *Given a fixed u_L, A learning algorithm $\mathcal{A}'$* asymptotically Pareto-dominates *a learning algorithm $\mathcal{A}$ if for all optimizer payoffs u_O, $V_L(\mathcal{A}', u_O) \geq V_L(\mathcal{A}, u_O)$, and for a positive measure set of optimizer payoffs u_O, $V_L(\mathcal{A}', u_O) > V_L(\mathcal{A}, u_O)$. A learning algorithm $\mathcal{A}$ is* asymptotically Pareto-optimal *if it is not asymptotically Pareto-dominated by any learning algorithm.*

Classes of learning algorithms. We will be interested in three specific classes of learning algorithms: *no-regret algorithms*, *no-swap-regret algorithms*, and *mean-based algorithms* (along with their subclass of *FTRL algorithms*).

A learning algorithm $\mathcal{A}$ is a *no-regret algorithm* if it is the case that, regardless of the sequence of actions $(x_1, x_2, \ldots, x_T)$ taken by the optimizer, the learner's utility satisfies:

$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \left(\max_{y^*\in[n]} \sum_{t=1}^{T} u_L(x_t, y^*)\right) - o(T).$$

A learning algorithm $\mathcal{A}$ is a *no-swap-regret algorithm* if it is the case that, regardless of the sequence of actions $(x_1, x_2, \ldots, x_T)$ taken by the optimizer, the learner's utility satisfies:

$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \max_{\pi:[n]\to[n]} \sum_{t=1}^{T} u_L(x_t, \pi(y_t)) - o(T).$$

Here the maximum is over all swap functions $\pi : [n] \to [n]$ (extended linearly to act on elements y_t of Δ_n). It is a fundamental result in the theory of online learning that both no-swap-regret algorithms and no-regret algorithms exist (see [Cesa-Bianchi and Lugosi, 2006]).

Some no-regret algorithms have the property that each round, they approximately best-respond to the historical sequence of losses. Following [Braverman et al., 2018] and [Deng et al., 2019b], we call such algorithms *mean-based algorithms*. Formally, we define mean-based algorithms as follows.

Definition 2.2. *A learning algorithm $\mathcal{A}$ is $\gamma(t)$-*mean-based *if whenever $j, j' \in [m]$ satisfy*

$$\frac{1}{t}\sum_{s=1}^{t} u_L(x_t, j') - \frac{1}{t}\sum_{s=1}^{t} u_L(x_s, j) \geq \gamma(t),$$

then $y_{t,j} \le \gamma(t)$ (i.e., if j is at least $\gamma(t)$ worse than some other action j' against the historical average action of the opponent, then the total probability weight on j must be at most $\gamma(t)$). A learning algorithm is mean-based *if it is $\gamma(t)$-mean-based for some $\gamma(t) = o(1)$.*

Many standard no-regret learning algorithms are mean-based, including Multiplicative Weights, Hedge, Online Gradient Descent, and others (see [Braverman et al., 2018]). In fact, all of the aforementioned algorithms can be viewed as specific instantiations of the mean-based algorithm Follow-The-Regularized-Leader. It is this subclass of mean-based algorithms that we will eventually show is Pareto-dominated in Section B; we define it below.

Definition 2.3. *$\mathit{FTRL}_T(\eta, R)$ is the horizon-dependent algorithm for a given learning rate $\eta > 0$ and bounded, strongly convex regularizer $R : \Delta^n \to \mathbb{R}$ which picks action $y_t \in \Delta^n$ via $y_t = \arg\max_{y \in \Delta^n} \left(\sum_{s=1}^{t-1} u_L(x_s, y) - \frac{R(y)}{\eta} \right)$. A learning algorithm $\mathcal{A}$ belongs to the family of learning algorithms FTRL if for all $T > 0$, the finite-horizon A_T is of the form $\mathit{FTRL}_T(\eta_T, R)$ for some sequence of learning rates η_T with $\eta_T = 1/o(T)$ and fixed regularizer R.*

As mentioned, the family FTRL contains many well-known algorithms. For instance, we can recover Multiplicative Weights with the negative entropy regularizer $R_T(y) = \sum_{i \in [n]} y_i \log y_i$, and Online Gradient Descent via the quadratic regularizer $R_T(y) = \frac{1}{2}||y||_2^2$ (see [Hazan, 2012] for details).

Other game-theoretic preliminaries and assumptions. Fix a specific game G. For any mixed strategy x of the optimizer, let $\mathrm{BR}_L(x) = \arg\max_{y \in [n]} u_L(x, y)$ represent the set of best-responses to x for the learner. Similarly, define $\mathrm{BR}_O(y) = \arg\max_{x \in [m]} u_O(x, y)$.

A *correlated strategy profile (CSP)* φ is an element of Δ_{mn} and represents a correlated distribution over pairs $(i, j) \in [m] \times [n]$ of optimizer/learner actions. For each $i \in [m]$ and $j \in [n]$, φ_{ij} represents the probability that the optimizer plays i and the learner plays j under φ. For mixed strategies $x \in \Delta_m$ and $y \in \Delta_n$, we will use tensor product notation $x \otimes y$ to denote the CSP corresponding to the product distribution of x and y. We also extend the definitions of u_L and u_O to CSPs (via $u_L(\varphi) = \sum_{i,j} \varphi_{ij} u_L(i, j)$, and likewise for $u_O(\varphi)$).

Throughout the rest of the paper, we will impose two constraints on the set of games G we consider (really, on the learner payoffs u_L we consider). These constraints serve the purpose of streamlining the technical exposition of our results, and both constraints only remove a measure-zero set of games from consideration. The first constraint is that we assume that over all possible strategy profiles, there is one which is uniquely optimal for the learner; i.e., a pair of moves $i^* \in [m]$ and $j^* \in [n]$ such that $u_L(i^*, j^*) > u_L(i, j)$ for any $(i, j) \neq (i^*, j^*)$. Note that slightly perturbing the entries of any payoff u_L causes this to be true with probability 1. We let $\varphi^+ = (i^*) \otimes (j^*)$ denote the corresponding optimal CSP.

Secondly, we assume that the learner has no weakly dominated actions. To define this, we say an action $y \in [n]$ for the learner is *strictly dominated* if it is impossible for the optimizer to incentivize y; i.e., there doesn't exist any $x \in \Delta_m$ for which $y \in \mathrm{BR}_L(x)$. We say an action $y \in [n]$ for the learner is *weakly dominated* if it is *not* strictly dominated but it is impossible for the optimizer to *uniquely* incentivize y; i.e., there doesn't exist any $x \in \Delta_m$ for which $\mathrm{BR}_L(x) = \{y\}$. Note that this is solely a constraint on u_L (not on u_O) and that we still allow for the possibility of the learner having strictly dominated actions. Moreover, only a measure-zero subset of possible u_L contain weakly dominated actions, since slightly perturbing the utilities of a weakly dominated action causes it to become either strictly dominated or non-dominated. This constraint allows us to remove some potential degeneracies (such as the learner having multiple copies of the same action) which in turn simplifies the statement of some results (e.g., Theorem 3.9).

Game-agnostic learning algorithms. Here we have defined learning algorithms as being associated with a fixed u_L and being able to observe the optimizer's sequence of actions (if not their actual payoffs). However many natural learning algorithms (including Multiplicative Weights and FTRL) only require the counterfactual payoffs of each action from each round. In the full version [Arunachaleswaran et al., 2024] we explore Pareto-optimality and Pareto-domination over this class of algorithms.

3 From learning algorithms to menus

3.1 The asymptotic menu of a learning algorithm

Our eventual goal is to understand which learning algorithms are Pareto-optimal for the learner. However, learning algorithms are fairly complex objects; instead, we will show that for our purposes we can associate each learning algorithm with a much simpler object we call an *asymptotic menu*, which can be represented as a convex subset of Δ_{mn}. Intuitively, the asymptotic menu of a learning algorithm describes the set of correlated strategy profiles an optimizer can asymptotically incentivize in the limit as T approaches infinity.

More formally, for a fixed horizon-dependent algorithm A^T, define the *menu* $\mathcal{M}(A^T) \subseteq \Delta_{mn}$ of A^T to be the convex hull of all CSPs of the form $\frac{1}{T}\sum_{t=1}^{T} x_t \otimes y_t$, where $(x_1, x_2, \ldots, x_T)$ is any sequence of optimizer actions and $(y_1, y_2, \ldots, y_T)$ is the response of the learner to this sequence under A^T (i.e., $y_t = A_t^T(x_1, x_2, \ldots, x_{t-1})$).

If a learning algorithm $\mathcal{A}$ has the property that the sequence $\mathcal{M}(A^1), \mathcal{M}(A^2), \ldots$ converges under the Hausdorff metric[1], we say that the algorithm $\mathcal{A}$ is *consistent* and call this limit value the *asymptotic menu* $\mathcal{M}(\mathcal{A})$ of $\mathcal{A}$. More generally, we will say that a subset $\mathcal{M} \subseteq \Delta_{mn}$ is an asymptotic menu if it is the asymptotic menu of some consistent algorithm. It is possible to construct learning algorithms that are not consistent (for example, imagine an algorithm that runs multiplicative weights when T is even, and always plays action 1 when T is odd); however even in this case we can find subsequences of time horizons where this converges and define a reasonable notion of asymptotic menu for such algorithms. We defer discussion of this to the full version [Arunachaleswaran et al., 2024], and otherwise will only concern ourselves with consistent algorithms. See also the full version for some explicit examples of asymptotic menus.

The above definition of asymptotic menu allows us to recast the Stackelberg game played by the learner and optimizer in more geometric terms. Given some u_L, the learner begins by picking a valid asymptotic menu $\mathcal{M}$. The optimizer then picks a point φ on $\mathcal{M}$ that maximizes $u_O(\varphi)$ (breaking ties in favor of the learner). The optimizer and the learner then receive utility $u_O(\varphi)$ and $u_L(\varphi)$ respectively.

For any asymptotic menu $\mathcal{M}$, define $V_L(\mathcal{M}, u_O)$ to be the utility the learner ends up with under this process. Specifically, define $V_L(\mathcal{M}, u_O) = \max\{u_L(\varphi) \mid \varphi \in \arg\max_{\varphi \in \mathcal{M}} u_O(\varphi)\}$. We can verify that this definition is compatible with our previous definition of V_L as a function of the learning algorithm $\mathcal{A}$ (see the full version [Arunachaleswaran et al., 2024] for proof).

Lemma 3.1. *For any learning algorithm $\mathcal{A}$, $V_L(\mathcal{M}(\mathcal{A}), u_O) = V_L(\mathcal{A}, u_O)$.*

As a consequence of Lemma 3.1, instead of working with asymptotic Pareto-dominance of learning algorithms, we can entirely work with Pareto-dominance of *asymptotic menus*, defined as follows.

Definition 3.2 (Pareto-dominance for asymptotic menus). *Fix a payoff u_L for the learner. An asymptotic menu $\mathcal{M}'$* Pareto-dominates *an asymptotic menu $\mathcal{M}$ if for all optimizer payoffs u_O,*

[1]The *Hausdorff distance* between two bounded subsets X and Y of Euclidean space is given by $d_H(X, Y) = \max(\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(y, X))$, where $d(a, B)$ is the minimum Euclidean distance between point a and the set B.

$V_L(\mathcal{M}, u_O) \geq V_L(\mathcal{M}', u_O)$, and for at least one[2] payoff u_O, $V_L(\mathcal{M}, u_O) > V_L(\mathcal{M}', u_O)$. An asymptotic menu $\mathcal{M}$ is Pareto-optimal *if it is not Pareto-dominated by any asymptotic menu.*

3.2 Characterizing possible asymptotic menus

Before we address the harder question of which asymptotic menus are Pareto-optimal, it is natural to wonder which asymptotic menus are even possible: that is, which convex subsets of Δ_{mn} are even attainable as asymptotic menus of some learning algorithm. In this section we provide a complete characterization of all possible asymptotic menus, which we describe below.

Theorem 3.3. *A closed, convex subset $\mathcal{M} \subseteq \Delta_{mn}$ is an asymptotic menu iff for every $x \in \Delta_m$, there exists a $y \in \Delta_n$ such that $x \otimes y \in \mathcal{M}$.*

The necessity condition of Theorem 3.3 follows quite straightforwardly from the observation that if the optimizer only ever plays a fixed mixed strategy $x \in \Delta_m$, the resulting average CSP will be of the form $x \otimes y$ for some $y \in \Delta_n$. The trickier part is proving sufficiency. For this, we will need to rely on the following two lemmas.

The first lemma applies Blackwell approachability to show that any $\mathcal{M}$ of the form specified in Theorem 3.3 must *contain* a valid asymptotic menu.

Lemma 3.4. *Assume the closed convex set $\mathcal{M} \subseteq \Delta_{mn}$ has the property that for every $x \in \Delta_m$, there exists a $y \in \Delta_n$ such that $x \otimes y \in \mathcal{M}$. Then there exists an asymptotic menu $\mathcal{M}' \subseteq \mathcal{M}$.*

Proof. We will show the existence of an algorithm $\mathcal{A}$ for which $\mathcal{M}(\mathcal{A}) \subseteq \mathcal{M}$. To do so, we will apply the Blackwell Approachability Theorem ([Blackwell, 1956]).

Consider the repeated vector-valued game in which the learner chooses a distribution $y_t \in \Delta_n$ over their n actions, the optimizer chooses a distribution $x_t \in \Delta_m$ over their m actions, and the learner receives the vector-valued, bilinear payoff $u(x_t, y_t) = x_t \otimes y_t$ (i.e., the CSP corresponding to this round). The Blackwell Approachability Theorem states that if the set $\mathcal{M}$ is response-satisfiable w.r.t. u – that is, for all $x \in \Delta_m$, there exists a $y \in \Delta_n$ such that $u(x, y) \in \mathcal{M}$ – then there exists a learning algorithm $\mathcal{A}$ such that

$$\lim_{T \to \infty} d\left(\frac{1}{T}\sum_{t=1}^{T} u(x_t, y_t), \mathcal{M}\right) = 0,$$

for any sequence of optimizer actions $\{x_t\}$ (here $d(p, S)$ represents the minimal Euclidean distance from point p to the set S). In words, the history-averaged CSP of play must approach to the set $\mathcal{M}$ as the time horizon grows. Since any $\varphi \in \mathcal{M}(\mathcal{A})$ can be written as the limit of such history-averaged payoffs (as $T \to \infty$), this would imply $\mathcal{M}(\mathcal{A}) \subseteq \mathcal{M}$.

Therefore all that remains is to prove that $\mathcal{M}$ is response-satisfiable. But this is exactly the property we assumed $\mathcal{M}$ to have, and therefore our proof is complete. □

The second lemma shows that asymptotic menus are *upwards closed*: if $\mathcal{M}$ is an asymptotic menu, then so is any convex set containing it.

Lemma 3.5. *If $\mathcal{M}$ is an asymptotic menu, then any closed convex set $\mathcal{M}'$ satisfying $\mathcal{M} \subseteq \mathcal{M}' \subseteq \Delta_{mn}$ is an asymptotic menu.*

[2]Alternatively, we can ask that one menu strictly beats the other on a positive measure set of payoffs. This may seem more robust, but turns out to be equivalent to the single-point definition. We prove this in the full version [Arunachaleswaran et al., 2024]. Note that by Lemma 3.1, this implies a similar equivalence for Pareto-domination of algorithms.

Proof Sketch. We defer the details of the proof to the full version [Arunachaleswaran et al., 2024] and provide a high-level sketch here. Since $\mathcal{M}$ is an asymptotic menu, we know there exists a learning algorithm $\mathcal{A}$ with $\mathcal{M}(\mathcal{A}) = \mathcal{M}$. We show how to take $\mathcal{A}$ and transform it to a learning algorithm $\mathcal{A}'$ with $\mathcal{M}(\mathcal{A}') = \mathcal{M}'$. The algorithm $\mathcal{A}'$ works as follows:

(1) At the beginning, the optimizer selects a point $\varphi \in \mathcal{M}'$ they want to converge to. They also agree on a "schedule" of moves (x_t, y_t) for both players to play whose history-average converges to the point φ without ever leaving $\mathcal{M}'$. (The optimizer can communicate this to the learner solely through the actions they take in some sublinear prefix of the game – see the full proof for details).
(2) The learner and optimizer then follow this schedule of moves (the learner playing x_t and the optimizer playing y_t at round t). If the optimizer never defects, they converge to the point φ.
(3) If the optimizer ever defects from their sequence of play, the learner switches to playing the original algorithm $\mathcal{A}$. In the remainder of the rounds, the time-averaged CSP is guaranteed to converge to some point $\varphi_{\text{suff}} = \mathcal{M}(\mathcal{A}) = \mathcal{M}$. Since the time-averaged CSP of the prefix φ_{pre} lies in $\mathcal{M}'$, the overall time-averaged CSP will still lie in $\mathcal{M}'$, so the optimizer cannot incentivize any point outside of $\mathcal{M}'$.

□

Combining Lemmas 3.4 and Lemmas 3.5, we can now prove Theorem 3.3.

Proof of Theorem 3.3. As mentioned earlier, the necessity condition is straightforward: assume for contradiction that there exists an algorithm $\mathcal{A}$ with asymptotic menu $\mathcal{M}$ such that, for some $x \in \Delta_m$, there is no point in $\mathcal{M}$ of the form $x \otimes y$ for any y. Then, let the optimizer play x in each round. The resulting CSP induced against $\mathcal{A}$ must be of the form $x \otimes y$ for some $y \in \Delta_n$, deriving a contradiction.

Now we will prove that if a set $\mathcal{M}$ has the property that $\forall x \in \Delta_m$, there exists a $y \in \Delta_n$ such that $x \otimes y \in \mathcal{M}$, then it is a valid menu. To see this, consider any set $\mathcal{M}$ with this property. Then by Lemma 3.4 there exists a valid menu $\mathcal{M}' \subseteq \mathcal{M}$. Then, by the upwards-closedness property of Lemma 3.5, the set $\mathcal{M} \supseteq \mathcal{M}'$ is also a menu.

□

3.3 No-regret and no-swap-regret menus

Another nice property of working with asymptotic menus is that no-regret and no-swap-regret properties of algorithms translate directly to similar properties on these algorithms' asymptotic menus (the situation for mean-based algorithms is a little bit more complex, and we discuss it in Section B).

To elaborate, say that the CSP φ is *no-regret* if it satisfies the no-regret constraint

$$\sum_{i \in [m]} \sum_{j \in [n]} \varphi_{ij} u_L(i, j) \geq \max_{j^* \in [n]} \sum_{i \in [m]} \sum_{j \in [n]} \varphi_{ij} u_L(i, j^*). \tag{2}$$

Similarly, say that the CSP φ is *no-swap-regret* if, for each $j \in [n]$, it satisfies

$$\sum_{i \in [m]} \varphi_{ij} u_L(i, j) \geq \max_{j^* \in [n]} \sum_{i \in [m]} \varphi_{ij} u_L(i, j^*). \tag{3}$$

For a fixed u_L, we will define the *no-regret menu* $\mathcal{M}_{NR}$ to be the convex hull of all no-regret CSPs, and the *no-swap-regret menu* $\mathcal{M}_{NSR}$ to be the convex hull of all no-swap-regret CSPs. In the following theorem we show that the asymptotic menu of any no-(swap-)regret algorithm is contained in the no-(swap-)regret menu.

Theorem 3.6. *If a learning algorithm $\mathcal{A}$ is no-regret, then for every u_L, $\mathcal{M}(\mathcal{A}) \subseteq \mathcal{M}_{NR}$. If $\mathcal{A}$ is no-swap-regret, then for every u_L, $\mathcal{M}(\mathcal{A}) \subseteq \mathcal{M}_{NSR}$.*

Note that both $\mathcal{M}_{NR}$ and $\mathcal{M}_{NSR}$ themselves are valid asymptotic menus, since for any $x \in \Delta_m$, they will contain some point of the form $x \otimes y$ for some $y \in \mathrm{BR}_L(x)$. In fact, we can say something much stronger about the no-swap-regret menu: it is exactly the convex hull of all such points.

Lemma 3.7. *The no-swap-regret menu $\mathcal{M}_{NSR}$ is the convex hull of all CSPs of the form $x \otimes y$, with $x \in \Delta_m$ and $y \in \mathrm{BR}_L(x)$.*

Proof. First, note that every CSP of the form $x \otimes y$, with $x \in \Delta_m$ and $y \in \mathrm{BR}_L(x)$, is contained in $\mathcal{M}_{NSR}$. This follows directly follows from the fact that this CSP satisfies the no-swap-regret constraint (3), since no action can be a better response than y to x.

For the other direction, consider a CSP $\varphi \in \mathcal{M}_{NSR}$. We will rewrite φ as a convex combination of product CSPs of the above form. For each pure strategy $a \in [n]$ for the learner, let $\beta(a) \in \Delta_m$ represent the conditional mixed strategy of the optimizer corresponding to X given that the learner plays action a, i.e. $\beta_j(a) = \frac{\varphi_{ja}}{\sum_{k \in [m]} \varphi_{ka}}$ for all $j \in [m]$ (setting $\beta_k(a)$ arbitrarily if all values φ_{ka} are zero). With this, we can write $\varphi = \sum_{a \in [m]} (\sum_{k \in [m]} \varphi_{ka})(\beta(a) \otimes a)$.

Now, note that if $a \notin \arg\max_b u_L(\beta(a), b)$, this would violate the no-swap-regret constraint (3) for $j = a$. Thus, we have rewritten φ as a convex combination of CSPs of the desired form, completing the proof. □

One key consequence of this characterization is that it allows us to show that the asymptotic menu of *any* no-regret algorithm must contain the no-swap-regret menu $\mathcal{M}_{NSR}$ as a subset. Intuitively, this is since every no-regret menu should also contain every CSP of the form $x \otimes y$ with $y \in \mathrm{BR}_L(x)$, since if the optimizer only plays x, the learner should learn to best-respond with y (although some care needs to be taken with ties).

Lemma 3.8. *For any no-regret algorithm $\mathcal{A}$, $\mathcal{M}_{NSR} \subseteq \mathcal{M}(\mathcal{A})$.*

This fact allows us to prove our first main result: that all consistent[3] no-swap-regret algorithms have the same asymptotic menu (namely, $\mathcal{M}_{NSR}$).

Theorem 3.9. *If $\mathcal{A}$ is a no-swap-regret algorithm, then $\mathcal{M}(\mathcal{A}) = \mathcal{M}_{NSR}$.*

Proof. From Theorem 3.6, $\mathcal{M}(A) \subseteq \mathcal{M}_{NSR}$. However, since any no-swap-regret algorithm also has no-regret, Lemma 3.8 implies $\mathcal{M}_{NSR} \subseteq \mathcal{M}(A)$. The conclusion follows. □

Note that in the proof of Theorem 3.9, we have appealed to Lemma 3.8 which uses the fact that u_L has no weakly dominated actions. This is necessary: consider, for example, a game with two identical actions for the learner, a and a' ($u_L(\cdot, a) = u_L(\cdot, a')$). We can consider two no-swap-regret algorithms for the learner, one which only plays a and never plays a', and the other which only plays a' and never plays a. These two algorithms will have different asymptotic menus, both of which contain only no-swap-regret CSPs. But as mentioned earlier, this is in some sense a degeneracy – the set of learner payoffs u_L with weakly dominated actions has zero measure (any small perturbation to u_L will prevent this from taking place).

Theorem 3.9 has a number of conceptual implications for thinking about learning algorithms in games:

[3]Actually, as a consequence of this result, it is possible to show that any no-swap-regret algorithm must be consistent: see the full version [Arunachaleswaran et al., 2024] for details.

(1) First, all no-swap-regret algorithms are *asymptotically equivalent*, in the sense that regardless of which no-swap-regret algorithm you run, any asymptotic strategy profile you converge to under one algorithm you could also converge to under another algorithm (for appropriate play of the other player). This is true even when the no-swap-regret algorithms appear qualitatively quite different in terms of the strategies they choose (compare e.g. the fixed-point based algorithm of [Blum and Mansour, 2007] with the more recent algorithms of [Dagan et al., 2023] and [Peng and Rubinstein, 2023]).
(2) In particular, there is no notion of regret that is meaningfully *stronger* than no-swap-regret for learning in (standard, normal-form) games. That is, there is no regret-guarantee you can feasibly insist on that would rule out some points of the no-swap-regret menu while remaining no-regret in the standard sense. In other words, the no-swap-regret menu is *minimal* among all no-regret menus: every no-regret menu contains $\mathcal{M}_{NSR}$, and no asymptotic menu (whether it is no-regret or not) is a subset of $\mathcal{M}_{NSR}$.
(3) Finally, these claims are *not* generally true for external regret. There are different no-regret algorithms with very different asymptotic menus (as a concrete example, $\mathcal{M}_{NR}$ and $\mathcal{M}_{NSR}$ are often different, and they are both asymptotic menus of some learning algorithm by Theorem 3.3).

Of course, this does not tell us whether it is actually *good* for the learner to use a no-swap-regret algorithm, from the point of view of the learner's utility. In the next section we will revisit this question through the lens of understanding which menus are Pareto optimal.

4 Characterizing Pareto-optimal menus

In this section we shift our attention to understanding which asymptotic menus are Pareto-optimal and which are Pareto-dominated by other asymptotic menus. The ideal result would be a characterization of all Pareto-optimal asymptotic menus; we will stop a little short of this and instead provide a full characterization of all Pareto-optimal *no-regret* menus that are also *polytopal* – i.e., can be written as the intersection of a finite number of half-spaces. This characterization will be sufficient for proving our main results that the no-swap-regret menu $\mathcal{M}_{NSR}$ is Pareto-optimal, but that the menu corresponding to multiplicative weights is sometimes Pareto-dominated.

Before we introduce the characterization, we introduce a little bit of additional notation. For any menu $\mathcal{M}$, let $U^+(\mathcal{M}) = \max_{\varphi \in \mathcal{M}} u_L(\varphi)$ denote the maximum learner payoff of any CSP in $\mathcal{M}$; likewise, define $U^-(\mathcal{M}) = \min_{\varphi \in \mathcal{M}} u_L(\varphi)$. We will also let $\mathcal{M}^+ = \arg\max_{\varphi \in \mathcal{M}} u_L(\varphi)$ and $\mathcal{M}^- = \arg\min_{\varphi \in \mathcal{M}} u_L(\varphi)$ be the subsets of $\mathcal{M}$ that attain this maximum and minimum (we will call these the *maximum-value* and *minimum-value sets* of $\mathcal{M}$).

Our characterization can now be simply stated as follows.

Theorem 4.1. *Let $\mathcal{M}$ be a polytopal no-regret menu. Then $\mathcal{M}$ is Pareto-optimal iff $\mathcal{M}^- = \mathcal{M}^-_{NSR}$. That is, $\mathcal{M}$ must share the same minimum-value set as the no-swap-regret menu $\mathcal{M}_{NSR}$.*

Note that while this characterization only allows us to reason about the Pareto-optimality of polytopal no-regret menus, in stating that these menus are Pareto-optimal, we are comparing them to all possible asymptotic menus. That is, we show that they are not Pareto-dominated by any possible asymptotic menu, even one which may have high regret and/or be an arbitrary convex set. We conjecture that this characterization holds for all no-regret menus (even ones that are not polytopal).

The remainder of this section will be dedicated to proving Theorem 4.1. We will begin in Section 4.1 by establishing some basic properties about $\mathcal{M}^+$, $\mathcal{M}^-$, $U^-(\mathcal{M})$, and $U^+(\mathcal{M})$ for no-regret and Pareto-optimal menus. Then in Section 4.2 we prove our main technical lemma (Lemma 4.4), which shows that a menu cannot be Pareto-dominated by a menu with a larger minimal set.

Finally, we complete the proof of Theorem 4.1 in Section 4.3, and discuss some implications for the Pareto-optimality of the no-regret and no-swap-regret menus in Section A.

4.1 Constraints on learner utilities

We begin with some simple observations on the possible utilities of the learner under Pareto-optimal menus and no-regret menus. We first consider $\mathcal{M}^+$. Recall that (by assumption) there is a unique pure strategy profile $\varphi^+ = (i^*) \otimes (j^*)$ that maximizes the learner's reward. We claim that any Pareto-optimal menu must contain φ^+.

Lemma 4.2. *If $\mathcal{M}$ is a Pareto-optimal asymptotic menu, then $\mathcal{M}^+ = \{\varphi^+\}$.*

Proof. Assume $\mathcal{M}$ is a Pareto-optimal asymptotic menu that does not contain φ^+. By Lemma 3.5, the set $\mathcal{M}' = \text{conv}(\mathcal{M}, \varphi^+)$ is also a valid asympotic menu. We claim $\mathcal{M}'$ Pareto-dominates $\mathcal{M}$.

To see this, first note that when $u_O = u_L$, $V_L(\mathcal{M}', u_O) = u_L(\varphi^+) > V_L(\mathcal{M}, u_O)$, since φ^+ is the unique CSP in Δ_{mn} maximizing u_L. On other hand, for any other u_O, the maximizer of u_O over $\mathcal{M}'$ is either equal to the maximizer of u_O over $\mathcal{M}$, or equal to φ^+. In either case, the learner's utility is at least as large, so $V_L(\mathcal{M}', u_O) \geq V_L(\mathcal{M}, u_O)$ for all u_O. It follows that $\mathcal{M}'$ Pareto-dominates $\mathcal{M}$. □

Note also that φ^+ belongs to $\mathcal{M}_{NSR}$ (since it a best-response CSP of the same form as in Lemma 3.7), so $\mathcal{M}^+_{NSR} = \varphi^+$. Since $\mathcal{M}_{NSR}$ is also contained in every no-regret menu, this also means that for any (not necessarily Pareto-optimal) no-regret menu $\mathcal{M}$, $\mathcal{M}^+ = \mathcal{M}^+_{NSR} = \varphi^+$.

We now consider the minimum-value set $\mathcal{M}^-$. Unlike for $\mathcal{M}^+$, it is no longer the case that all Pareto-optimal menus share the same set $\mathcal{M}^-$. It is not even the case (as we shall see in Section 4.3), that all Pareto-optimal menus have the same minimum learner utility $U^-(\mathcal{M})$.

However, it is the case that all *no-regret* algorithms share the same value for the minimum learner utility $U^-(\mathcal{M})$, namely the "zero-sum" utility $U_{ZS} = \min_{x \in \Delta_m} \max_{y \in \Delta_n} u_L(x, y)$. The reason for this is that U_{ZS} is the largest utility the learner can guarantee when playing a zero-sum game (i.e., when the optimizer has payoffs $u_O = -u_L$), and thus it is impossible to obtain a higher value of $U^-(\mathcal{M})$. This is formalized in the following lemma.

Lemma 4.3. *Every asymptotic menu must have $U^-(\mathcal{M}) \leq U_{ZS}$. Moreover, if $\mathcal{M}$ is a no-regret asymptotic menu, then $U^-(\mathcal{M}) = U_{ZS}$, and $\mathcal{M}^-_{NSR} \subseteq \mathcal{M}^-$.*

Proof. Let (x_{ZS}, y_{ZS}) be the solution to the minimax problem $\min_{x \in \Delta_m} \max_{y \in \Delta_n} u_L(x, y)$ (i.e., the Nash equilibrium of the corresponding zero-sum game). By Theorem 3.3, any asymptotic menu $\mathcal{M}$ must contain a point of the form $x_{ZS} \otimes y$. By construction, $u_L(x_{ZS} \otimes y) \leq U_{ZS}$, so $U^-(\mathcal{M}) \leq U_{ZS}$.

To see that every no-regret asymptotic menu satisfies $U^-(\mathcal{M}) = U_{ZS}$, assume that $\mathcal{M}$ is a no-regret menu, and $\varphi \in \mathcal{M}$ satisfies $u_L(\varphi) < U_{ZS}$. Since φ has no-regret (satisfies the conditions of (2)), we must also have $u_L(\varphi) \geq \max_{y \in \Delta_n} \min_{x \in \Delta_m} u_L(x, y)$, since this holds for whatever marginal distribution x is played by the optimizer under φ. But by the minimax theorem, $\max_{y \in \Delta_n} \min_{x \in \Delta_m} u_L(x, y) = U_{ZS}$, and so we have a contradiction.

Finally, note that since $\mathcal{M}$ is no-regret, $\mathcal{M}_{NSR} \subseteq \mathcal{M}$ and so $\mathcal{M}^-_{NSR} \subseteq \mathcal{M}^-$ (since they share the same minimum value). □

4.2 Pareto-domination and minimum-value sets

We now present our two main lemmas necessary for the proof of Theorem 4.1. The first lemma shows that if one menu contains a point not present in the second menu (and both menus share the same maximum-value set), then the first menu cannot possibly Pareto-dominate the second menu.

Lemma 4.4. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two distinct asymptotic menus where $\mathcal{M}_1^+ = \mathcal{M}_2^+$. Then if either:*

- *i.* $\mathcal{M}_2 \setminus \mathcal{M}_1 \neq \emptyset$, *or*
- *ii.* $\mathcal{M}_1^- = \mathcal{M}_2^-$,

then there exists a u_O *for which* $V_L(\mathcal{M}_1, u_O) > V_L(\mathcal{M}_2, u_O)$ *(i.e.,* $\mathcal{M}_2$ *does not Pareto-dominate* $\mathcal{M}_1$*).*

Note that Lemma 4.4 holds also under the secondary assumption that $\mathcal{M}_1^- = \mathcal{M}_2^-$. One important consequence of this is that all menus with identical minimum value and maximum value sets $\mathcal{M}^-$ and $\mathcal{M}^+$ are incomparable to each other under the Pareto-dominance order (even such sets that may contain each other).

The key technical ingredient for proving Lemma 4.4 is the following lemma, which establishes a "two-dimensional" variant of the above claim.

Lemma 4.5. *Let* $f, g : [a, b] \to \mathbb{R}$ *be two distinct concave functions satisfying* $f(a) \leq g(a)$ *and* $f(b) = g(b)$*. For* $\theta \in [0, \pi]$*, let* $\hat{f}(\theta) = \arg\max_{x \in [a,b]} (x \cos\theta + f(x) \sin\theta)$ *(if the argmax is not unique, then* $\hat{f}(\theta)$ *is undefined). Define* $\hat{g}(\theta)$ *symmetrically. Then there exists a* θ *for which* $\hat{f}(\theta) > \hat{g}(\theta)$*.*

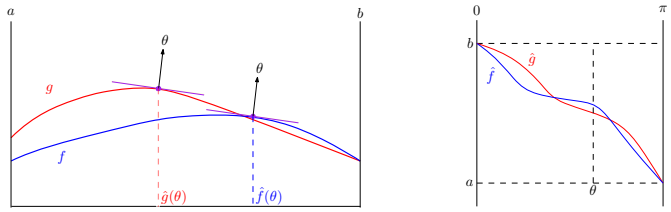

Fig. 1. A visual depiction of Lemma 4.5. The purple points in the left figure denote the maximizers of f and g in the direction θ; since the purple point on f is to the right of that on g, we have $\hat{f}(\theta) > \hat{g}(\theta)$ for this θ.

Proof. Since $f(x)$ is a concave curve, it has a (weakly) monotonically decreasing derivative $f'(x)$. This derivative is not necessarily defined for all $x \in [a, b]$, but since f is concave it is defined almost everywhere. At points c where it is not defined, f still has a well-defined left derivative $f'_L(x) = \lim_{h \to 0} (f(x) - f(x-h))/h$ and right derivative $f'_R(x) = \lim_{h \to 0} (f(x+h) - f(x))/h$. We will abuse notation and let $f'(x)$ denote the interval $[f'_L(x), f'_R(x)]$ (at the boundaries defining $f'(a) = (-\infty, f_R(a)]$ and $f'(b) = [f_L(b), \infty)$. Similarly, the interval-valued inverse function $(f')^{-1}(y)$ is also well-defined, decreasing in y, and uniquely-defined for almost all values of y in $(-\infty, \infty)$.

Note that since $\hat{f}(\theta)$ is the x coordinate of the point on the curve $f(x)$ that maximizes the inner product with the unit vector $(\cos\theta, \sin\theta)$, $f'(\hat{f}(\theta))$ must contain the value $-\cos\theta / \sin\theta = -\cot\theta$. In particular, if $\hat{f}(\theta)$ is uniquely defined, $\hat{f}(\theta) = (f')^{-1}(-\cot\theta)$. So it suffices to find a y for which $(f')^{-1}(y) > (g')^{-1}(y)$.

To do this, we make the following observation: since $f(b) - f(a) \geq g(b) - g(a)$, $\int_a^b f'(x)dx \geq \int_a^b g'(x)dx$ [4]. This means there must be a point $c \in (a, b)$ where $f'(c) > g'(c)$. If not, then we must have $f'(x) \leq g'(x)$ for all $x \in (a, b)$; but the only way we can simultaneously have $f'(x) \leq g'(x)$ for all $x \in (a, b)$ and $\int_a^b f'(x)dx \geq \int_a^b g'(x)dx$ is if $f'(x) = g'(x)$ for almost all $x \in (a, b)$ – but this would contradict the fact that f and g are distinct concave functions.

Now, take a point $c \in (a, b)$ where $f'(c) > g'(c)$ and choose a y in $f'(c)$. Since g' is a decreasing function, there must exist a $c' < c$ such that $y \in g'(c')$, and so $(f')^{-1}(y) > (g')^{-1}(y)$. □

[4]These integrals are well defined because the first derivatives of f and g exist almost everywhere.

We can now prove Lemmas 4.4 through an application of the above lemma.

Proof of Lemma 4.4. We will consider the two preconditions separately, and begin by considering the case where $\mathcal{M}_1^- = \mathcal{M}_2^-$. Since $\mathcal{M}_1$ and $\mathcal{M}_2$ are distinct asymptotic menus, there must be an extreme point φ in one menu that does not belong to the other. In particular, there must exist an optimizer payoff u_O where $u_O(\varphi) > u_O(\varphi')$ for any φ' in the other menu. Denote this specific optimizer payoff by u_O^0.

We will show that there exists a $u_O \in \text{span}(u_L, u_O^0)$ where $V_L(\mathcal{M}_1, u_O) > V_L(\mathcal{M}_2, u_O)$. To do this, we will project $\mathcal{M}_1$ and $\mathcal{M}_2$ to two-dimensional sets S_1 and S_2 by letting $S_1 = \{(u_L(\varphi), u_O^0(\varphi)) \mid \varphi \in \mathcal{M}_1\}$ (and defining S_2 symmetrically). By our construction of u_O^0, these two convex sets S_1 and S_2 are distinct. Also, note that if $u_O = \lambda_1 u_L + \lambda_2 u_O^0$, we can interpret $V_L(\mathcal{M}_1, u_O)$ as the maximum value of z_1 for any point in $\arg\max_{(z_1,z_2)\in S_1}(\lambda_1 z_1 + \lambda_2 z_2)$. We can interpret $V_L(\mathcal{M}_2, u_O)$ similarly.

Let us now consider the geometry of S_1 and S_2. Let u^- denote the common value of $U^-(\mathcal{M}_1)$ and $U^-(\mathcal{M}_2)$, and similarly, let u^+ denote the common value of $U^+(\mathcal{M}_1)$ and $U^+(\mathcal{M}_2)$. Both S_1 and S_2 are contained in the "vertical" strip $u^- \le z_1 \le u^+$. We can therefore write S_1 as the region between the concave curve $f_{\text{up}} : [u^-, u^+]$ (representing the upper convex hull of S_1) and f_{down} (representing the lower convex hull of S_1; define g_{up} and g_{down} analogously for S_2. Since S_1 and S_2 are distinct, either $f_{\text{up}} \neq g_{\text{up}}$ or $f_{\text{down}} \neq g_{\text{down}}$; without loss of generality, assume $f_{\text{up}} \neq g_{\text{up}}$ (we can switch the upper and lower curves by changing u_O^0 to $-u_O^0$).

Note also that since $\mathcal{M}_1^+ = \mathcal{M}_2^+$ and $\mathcal{M}_1^- \subseteq \mathcal{M}_2^-$, we have $f_{\text{up}}(u^+) = g_{\text{up}}(u^+)$ and $f_{\text{up}}(u^-) \le g_{\text{up}}(u^-)$ (since $[f_{\text{down}}(u^-), f_{\text{up}}(u^-)] \subseteq [g_{\text{down}}(u^-), g_{\text{up}}(u^-)]$). By Lemma 4.5, there exists a $\theta \in [0, \pi]$ for which $\hat{f}_{\text{up}}(\theta) > \hat{g}_{\text{up}}(\theta)$. But by the definition of $\hat{f}$ and $\hat{g}$, this implies that for $u_O = \cos(\theta)u_L + \sin(\theta)u_O^0$, $V_L(\mathcal{M}_1, u_O) > V_L(\mathcal{M}_2, u_O)$, as desired. This proof is visually depicted in Figure 1.

The remaining case, where $\mathcal{M}_2 \setminus \mathcal{M}_1 \neq \emptyset$, can be proved very similarly to the above proof. We make the following changes:

- First, we choose an extreme point φ of $\mathcal{M}_2$ that belongs to $\mathcal{M}_2$ but not $\mathcal{M}_1$. Again, we choose a u_O which separates φ from $\mathcal{M}_1$. We let $u^* = u_O(\varphi)$; note that $u^* < u^+$ (since $\mathcal{M}_1^+ = \mathcal{M}_2^+$).
- Instead of defining our functions f_{up} and g_{up} on the full interval $[u^-, u^+]$, we instead restrict them to the interval $[u^*, u^+]$. Because of our construction of φ, we have that $f_{up}(u^*) < g_{up}(u^*)$, and $f_{up}(u^+) = g_{up}(u^+)$.
- We can again apply Lemma 4.5 to these two functions on this sub-interval, and construct a u_O for which $V_L(\mathcal{M}_1, u_O) > V_L(\mathcal{M}_2, u_O)$.

□

One useful immediate corollary of Lemma 4.4 is that it is impossible for high-regret menus (menus that are not no-regret) to Pareto-dominate no-regret menus.

Corollary 4.6. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two asymptotic menus such that $\mathcal{M}_1$ is no-regret and $\mathcal{M}_2$ is* not *no-regret. Then $\mathcal{M}_2$ does not Pareto-dominate $\mathcal{M}_1$.*

Proof. If $\mathcal{M}_2$ does not contain φ^+, add it to $\mathcal{M}_2$ via Lemma 4.2 (this only increases the position of $\mathcal{M}_2$ in the Pareto-dominance partial order). Since $\mathcal{M}_1$ is no-regret, it must already contain φ^+, and therefore we can assume $\mathcal{M}_1^+ = \mathcal{M}_2^+ = \{\varphi^+\}$.

Since $\mathcal{M}_2$ is not no-regret, it must contain a CSP φ that does not lie in $\mathcal{M}_{NR}$, and therefore $\mathcal{M}_2 \setminus \mathcal{M}_1 \neq \emptyset$. It then follows from Lemma 4.4 that $\mathcal{M}_2$ does not Pareto-dominate $\mathcal{M}_1$. □

4.3 Completing the proof

We can now finish the proof of Theorem 4.1.

Proof of Theorem 4.1. We will first prove that if a no-regret menu $\mathcal{M}$ satisfies $\mathcal{M}^- = \mathcal{M}^-_{NSR}$, then it is Pareto-optimal. To do so, we will consider any other menu $\mathcal{M}'$ and show that $\mathcal{M}'$ does not Pareto-dominate $\mathcal{M}$. There are three cases to consider:

- **Case 1**: $U^-(\mathcal{M}') < U^-(\mathcal{M})$. In this case, $\mathcal{M}'$ cannot dominate $\mathcal{M}$ since $V_L(\mathcal{M}, -u_L) > V_L(\mathcal{M}', -u_L)$ (note that $U^-(\mathcal{M}) = V_L(\mathcal{M}, -u_L)$, since if $u_O = -u_L$, the optimizer picks the utility-minimizing point for the learner).
- **Case 2**: $U^-(\mathcal{M}') > U^-(\mathcal{M})$. By Lemma 4.3, this is not possible.
- **Case 3**: $U^-(\mathcal{M}') = U^-(\mathcal{M})$. If $\mathcal{M}'$ is not a no-regret menu, then by Corollary 4.6 it cannot dominate $\mathcal{M}$. We will therefore assume that $\mathcal{M}'$ is a no-regret menu, i.e. $\mathcal{M}' \subseteq \mathcal{M}_{NR}$ Then, by Lemma 4.3, $\mathcal{M}^- \subseteq (\mathcal{M}')^-$. Also, by Lemma 4.2, we can assume without loss of generality that $(\mathcal{M}')^+ = \{\varphi^+\} = \mathcal{M}^+$ (if $\mathcal{M}'$ does not contain φ^+, replace it with the Pareto-dominating menu that contains it). Now, by Lemma 4.4, $\mathcal{M}'$ does not dominate $\mathcal{M}$.

We now must show that if $\mathcal{M}^- \neq \mathcal{M}^-_{NSR}$, then it is Pareto-dominated by some other menu. Since $\mathcal{M}$ is (by assumption) a no-regret menu, we must have $U^-(\mathcal{M}) = U^-(\mathcal{M}_{NSR}) = U_{ZS}$, and $\mathcal{M}^- \supset \mathcal{M}^-_{NSR}$ (Lemmas 4.3). Consider an extreme point φ_0 that belongs to $\mathcal{M}^-$ but not to $\mathcal{M}^-_{NSR}$. Construct the menu $\mathcal{M}'$ as follows: it is the convex hull of $\mathcal{M}_{NSR}$ and all the extreme points in $\mathcal{M}$ *except for* φ_0. By Lemma 3.5, this is a valid menu (it is formed by adding some points to the valid menu $\mathcal{M}_{NSR}$). Note also that $\mathcal{M}'$ has all the same extreme points of $\mathcal{M}$ except for φ_0 (since $\mathcal{M}$ is a polytope, we add a finite number of extreme points to $\mathcal{M}'$, all of which are well-separated from φ_0), and in particular is distinct from $\mathcal{M}$.

We will show that $\mathcal{M}'$ Pareto-dominates $\mathcal{M}$. To see this, note first that, by Lemma 4.4, there is some u_O such that $V_L(U_{\mathcal{M}}, u_O) < V_L(U_{\mathcal{M}'}, u_O)$. Furthermore, for all other values of u_O, $V_L(U_{\mathcal{M}}, u_O) \leq V_L(U_{\mathcal{M}'}, u_O)$. This is since the maximizer of u_O over $\mathcal{M}$ is either the minimal-utility point φ_0 (which cannot be strictly better than the maximizer of u_O over $\mathcal{M}'$), or exactly the same point as the maximizer of u_O over $\mathcal{M}'$. It follows that $\mathcal{M}'$ Pareto-dominates $\mathcal{M}$. □

Note that in the Proof of Theorem 4.1, we only rely on the fact that the menu $\mathcal{M}$ is polytopal in precisely one spot, when we construct a menu $\mathcal{M}'$ that Pareto-dominates $\mathcal{M}$ by "removing" an extreme point from $\mathcal{M}^-$. As stated, this removal operation requires $\mathcal{M}$ to be a polytope: in general, it is possible that any extreme point φ_0 that belongs to $\mathcal{M}^-$ is a limit of other extreme points in $\mathcal{M}$, and so when attempting to construct $\mathcal{M}'$ per the procedure above, we would just perfectly recover the original $\mathcal{M}$ when taking the convex closure of the remaining points.

That said, it is not clear whether the characterization of non-polytopal Pareto-optimal menus is any different than the characterization in Theorem 4.1. In fact, by the argument in the proof of Theorem 4.1, one direction of the characterization still holds (if a non-polytopal no-regret menu satisfies $\mathcal{M}^- = \mathcal{M}^-_{NSR}$, then it is Pareto-optimal). We conjecture that this characterization holds for non-polytopal menus (and leave it as an interesting open problem).

Conjecture 4.7. *Any no-regret menu $\mathcal{M}$ is Pareto-optimal iff $\mathcal{M}^- = \mathcal{M}_{NSR}$.*

On the other hand, the restriction to no-regret menus is necessary for the characterization of Theorem 4.1 to hold. To see this, note that another interesting corollary of Lemma 4.4 is that any minimal asymptotic menu is Pareto-optimal (in fact, we have the slightly stronger result stated below).

Corollary 4.8. *Let $\mathcal{M}$ be an inclusion-minimal asymptotic menu (i.e., with the property that no other asymptotic menu $\mathcal{M}'$ satisfies $\mathcal{M}' \subset \mathcal{M}$). Then the menu $\mathcal{M}' = \text{conv}(\mathcal{M}, \{\varphi^+\})$ is Pareto-optimal.*

Corollary 4.8 allows us to construct some high-regret asymptotic menus that are Pareto-optimal. For example, we can show that the algorithm that always plays the learner's component of φ^+ is Pareto-optimal.

Theorem 4.9. *Let $\mathcal{M}$ be the asymptotic menu of the form $\mathcal{M} = \{x \otimes j^* \mid x \in \Delta_m\}$ (where j^* is the learner's component of $\varphi^+ = (i^*) \otimes (j^*)$). Then $\mathcal{M}$ is Pareto-optimal.*

Proof. By Theorem 3.3, $\mathcal{M}$ is inclusion-minimal. Since $\mathcal{M}$ also includes the CSP φ^+, it is Pareto-optimal by Corollary 4.8. □

Note that in general, the menu $\mathcal{M}$ in Theorem 4.9 is not no-regret, and may have $U^-(\mathcal{M}) < U_{ZS}$. We leave it as an interesting open question to provide a full characterization of all Pareto-optimal asymptotic menus.

In the interest of space, we discuss the implications of these results for no-regret and no-swap-regret menus in Appendix A.

Acknowledgments

This work is supported by NSF grants CCF-1910534 and CCF-2045128 and by an Amazon AWS Grant.

References

Ioannis Anagnostides, Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Noah Golowich, and Tuomas Sandholm. 2022a. Near-Optimal No-Regret Learning for Correlated Equilibria in Multi-Player General-Sum Games. In *ACM Symposium on Theory of Computing*.

Ioannis Anagnostides, Ioannis Panageas, Gabriele Farina, and Tuomas Sandholm. 2022b. On Last-Iterate Convergence Beyond Zero-Sum Games. In *International Conference on Machine Learning*.

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. 2024. Pareto-Optimal Algorithms for Learning in Games. arXiv:2402.09549

Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. 2015. Commitment Without Regrets: Online Learning in Stackelberg Security Games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (Portland, Oregon, USA) *(EC '15)*. Association for Computing Machinery, New York, NY, USA, 61–78. https://doi.org/10.1145/2764468.2764478

David Blackwell. 1956. An analog of the minimax theorem for vector payoffs. (1956).

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

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

William Brown, Jon Schneider, and Kiran Vodrahalli. 2023. Is Learning in Games Good for the Learners?. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=jR2FkqW6GB

Linda Cai, S Matthew Weinberg, Evan Wildenhain, and Shirley Zhang. 2023. Selling to multiple no-regret buyers. In *International Conference on Web and Internet Economics*. Springer, 113–129.

Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

Yiling Chen and Tao Lin. 2023. Persuading a Behavioral Agent: Approximately Best Responding and Learning. *arXiv preprint arXiv:2302.03719* (2023).

Natalie Collina, Eshwar Ram Arunachaleswaran, and Michael Kearns. 2023. Efficient Stackelberg Strategies for Finitely Repeated Games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 643–651.

Vincent Conitzer and Tuomas Sandholm. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*. 82–90.

Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. 2023. From External to Swap Regret 2.0: An Efficient Reduction and Oblivious Adversary for Large Action Spaces. *arXiv preprint arXiv:2310.19786* (2023).

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. 2019a. Prior-free dynamic auctions with low regret buyers. *Advances in Neural Information Processing Systems* 32 (2019).

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

Gabriele Farina, Ioannis Anagnostides, Haipeng Luo, Chung-Wei Lee, Christian Kroer, and Tuomas Sandholm. 2022. Near-Optimal No-Regret Learning Dynamics for General Convex Games. In *Neural Information Processing Systems (NeurIPS)*.

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

Drew Fudenberg and David K Levine. 1998. *The theory of learning in games*. Vol. 2. MIT press.

Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua R Wang, and S Matthew Weinberg. 2024. Contracting with a Learning Agent. *arXiv preprint arXiv:2401.16198* (2024).

Nika Haghtalab, Thodoris Lykouris, Sloan Nietert, and Alexander Wei. 2022. Learning in Stackelberg Games with Non-myopic Agents. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 917–918.

Sergiu Hart and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

Elad Hazan. 2012. 10 the convex optimization approach to regret minimization. *Optimization for machine learning* (2012), 287.

Yoav Kolumbus and Noam Nisan. 2022a. Auctions between regret-minimizing agents. In *Proceedings of the ACM Web Conference 2022*. 100–111.

Yoav Kolumbus and Noam Nisan. 2022b. How and why to manipulate your own agent: On the incentives of users of learning agents. *Advances in Neural Information Processing Systems* 35 (2022), 28080–28094.

Niklas Lauffer, Mahsa Ghasemi, Abolfazl Hashemi, Yagiz Savas, and Ufuk Topcu. 2022. No-Regret Learning in Dynamic Stackelberg Games. https://doi.org/10.48550/ARXIV.2202.04786

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

Janusz Marecki, Gerry Tesauro, and Richard Segal. 2012. Playing Repeated Stackelberg Games with Unknown Opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2* (Valencia, Spain) *(AAMAS '12)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 821–828.

Binghui Peng and Aviad Rubinstein. 2023. Fast swap regret minimization and applications to approximate correlated equilibria. *arXiv preprint arXiv:2310.19647* (2023).

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. 2019. Learning Optimal Strategies to Commit To. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 2149–2156. https://doi.org/10.1609/aaai.v33i01.33012149

Georgios Piliouras, Ryann Sim, and Stratis Skoulakis. 2022. Beyond Time-Average Convergence: Near-Optimal Uncoupled Online Learning via Clairvoyant Multiplicative Weights Update. *Advances in Neural Information Processing Systems* 35 (2022), 22258–22269.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. 2015. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems* 28 (2015).

H Peyton Young. 2004. *Strategic learning and its limits*. OUP Oxford.

Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen Marcus McAleer, Andreas Alexander Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. 2023. Computing Optimal Equilibria and Mechanisms via Learning in Zero-Sum Extensive-Form Games. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=yw1v4RqvPk

A Implications for no-regret and no-swap-regret menus

Already Theorem 4.1 has a number of immediate consequences for understanding the no-regret menu $\mathcal{M}_{NR}$ and the no-swap-regret menu $\mathcal{M}_{NSR}$, both of which are polytopal no-regret menus by their definitions in (2) and (3) respectively. As an immediate consequence of our characterization, we can see that the no-swap-regret menu (and hence any no-swap-regret learning algorithm) is Pareto-optimal.

Corollary A.1. *The no-swap-regret menu $\mathcal{M}_{NSR}$ is a Pareto-optimal asymptotic menu.*

It would perhaps be ideal if $\mathcal{M}_{NSR}$ was the unique Pareto-optimal no-regret menu, as it would provide a somewhat clear answer as to which learning algorithm one should use in a repeated game. Unfortunately, this is not the case – although $\mathcal{M}_{NSR}$ is the minimal Pareto-optimal no-regret menu, Theorem 4.1 implies there exist infinitely many distinct Pareto-optimal no-regret menus.

On the more positive side, Theorem 4.1 (combined with Lemma 3.5) gives a recipe for how to construct a generic Pareto-optimal no-regret learning algorithm: start with a no-swap-regret

learning algorithm (the menu $\mathcal{M}_{NSR}$) and augment it with any set of additional CSPs that the learner and optimizer can agree to reach. This can be any set of CSPs as long as i. each CSP φ has no regret, and ii. each CSP has learner utility $u_L(\varphi)$ strictly larger than the minimax value U_{ZS}.

Corollary A.2. *There exist infinitely many Pareto-optimal asymptotic menus.*

Finally, perhaps the most interesting consequence of Theorem 4.1 is that, despite this apparent wealth of Pareto-optimal menus and learning algorithms, the no-regret menu $\mathcal{M}_{NR}$ is very often *Pareto-dominated*. In particular, it is easy to find learner payoffs u_L for which $\mathcal{M}_{NR}^- \neq \mathcal{M}_{NSR}^-$, as we show below.

Corollary A.3. *There exists a learner payoff[5] u_L for which the no-regret $\mathcal{M}_{NR}$ is* not *a Pareto-optimal asymptotic menu.*

Proof. Take the learner's payoff from Rock-Paper-Scissors, where the learner and optimizer both have actions $\{a_1, a_2, a_3\}$, and $u_L(a_i, a_j) = 0$ if $j = i$, 1 if $j = i + 1 \bmod 3$, and -1 if $j = i - 1 \bmod 3$. For this game, $U_{ZS} = 0$ (the learner can guarantee payoff 0 by randomizing uniformly among their actions).

Now, note that the CSP $\varphi = (1/3)(a_1 \otimes a_1) + (1/3)(a_2 \otimes a_2) + (1/3)(a_3 \otimes a_3)$ has the property that $u_L(\varphi) = 0 = U_{ZS}$ and that $\varphi \in \mathcal{M}_{NR}$, but also that $\varphi \notin \mathcal{M}_{NSR}$ (e.g. it is beneficial for the learner to switch from playing a_1 to a_2). Since $\mathcal{M}_{NR}$ is a polytopal no-regret menu, it follows from our characterization in Theorem 4.1 that $\mathcal{M}_{NR}$ is not Pareto-optimal. □

B Mean-based algorithms and menus

In this section, we return to one of the main motivating questions of this work: are standard online learning algorithms (like multiplicative weights or follow-the-regularized-leader) Pareto-optimal? Specifically, are *mean-based* no-regret learning algorithms, which always approximately best-respond to the historical sequence of observed losses, Pareto-optimal?

We will show that the answer to this question is *no*: in particular, there exist payoffs u_L where the menus of some mean-based algorithms (specifically, menus for multiplicative weights and FTRL) are not Pareto-optimal. Our characterization of Pareto-optimal no-regret menus in the previous section (Theorem 4.1) does most of the heavy lifting here: it means that in order to show that a specific algorithm is not Pareto-optimal, we need only find a sequence of actions by the optimizer that both causes the learner to end up with the zero-sum utility u_{ZS} and high swap-regret (i.e., at a point not belonging to $\mathcal{M}^-$). Such games (and corresponding trajectories of play by the optimizer) are relatively easy to find – we will give one explicit example shortly that works for any mean-based algorithm.

However, there is a catch – our characterization in Theorem 4.1 only applies to *polytopal* menus (although we conjecture that it also applies to non-polytopal menus). So, in order to formally prove that a mean-based algorithm is not Pareto-optimal, we must additionally show that its corresponding menu is a polytope. Specifically, we give an example of a family of games where the asymptotic menus of all FTRL algorithms have a simple description as an explicit polytope which we can show is not Pareto-optimal.

Theorem B.1. *There exists a family of learner payoffs u_L with $m = 3$ actions for the learner and $n = 2$ actions for the optimizer where all FTRL algorithms are Pareto-dominated.*

In in the interest of space, the proof of this result has been deferred to the full version [Arunachaleswaran et al., 2024]. There, we introduce the concept of the "mean-based menu", a menu of CSPs that is

[5] In fact, there exists a positive measure of such u_L. It is easy to adapt this proof to work for small perturbations of the given u_L, see the full version [Arunachaleswaran et al., 2024] for a proof.

achievable against any mean-based algorithm, and introduce the family of games we study. We then show that for these games, the mean-based menu can be explicitly characterized as a polytope. Finally, we prove that the asymptotic menu of any instantiation of FTRL must actually equal this mean-based menu (instead of merely containing it).