The Learning Approach to Games

Melih İşeri[*] Erhan Bayraktar[†]

March 4, 2025

Abstract

This work provides a unified framework for exploring games. In existing literature, strategies of players are typically assigned scalar values, and the concept of Nash equilibrium is used to identify compatible strategies. However, this approach lacks the internal structure of a player, thereby failing to accurately model observed behaviors in reality. To address this limitation, we propose to characterize players by their learning algorithms, and as their estimations intrinsically induce a distribution over strategies, we introduced the notion of equilibrium in terms of characterizing the recurrent behaviors of the learning algorithms. This approach allows for a more nuanced understanding of players, and brings the focus to the challenge of learning that players face. While our explorations in discrete games, mean-field games, and reinforcement learning demonstrate the framework's broad applicability, they also set the stage for future research aimed at specific applications.

ACM Classification: I.2.6; J.4

[*] Department of Mathematics, University of Michigan, United States, iseri@umich.edu.

[†] Department of Mathematics, University of Michigan, United States, erhan@umich.edu.

1 Introduction

Game theory, like every branch of mathematics, explores fundamental concepts for systems that are as general as possible, where individual components have potential choices to make. The opportunities for exploration are vast and deeply complex, with implications across a diverse array of fields. These include societal structures, competitions in numerous games, various dynamics of businesses, computational decision processes, financial models, and many more. Recent advancements have even surpassed human capabilities in various domains, prompting increased efforts to better understand our brains, the most fascinating dynamic system.

Mathematically, we need to have fundamental concepts that are general enough to cover everything we aim to model. In the realm of games, we point out that what defines a player is a sequence of observations, a sequence of potential actions, and a collection of learning algorithms, all of which might be remarkably complicated. Given that the universe, or environment, is characterized by some abstract probability space $(\Omega^u, \mathcal{F}^u, \mathbb{P}^u)$, and denoting $\mathcal{P}(E)$ as the set of probability distributions on the set E, we introduce the definition of a player as arbitrary as possible;

Definition 1 *Let $\mathcal{E}$ be a space of observables, and $\mathscr{E}$ be the set of finite sequences of $\mathcal{E}$. Also, let $\mathbb{A}$ be a space of actions, and $\mathscr{A}$ be the set of finite sequences of $\mathbb{A}$. Moreover, let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be some arbitrary set of functions with domain $\mathcal{D}$, called spaces of estimations.*

We call $(\mathcal{O}, \mathfrak{L}_1, \cdots, \mathfrak{L}_k, \Upsilon)$ a player in the environment $(\Omega^u, \mathcal{F}^u, \mathbb{P}^u)$ with observations $\mathcal{O}$, learning algorithms $\mathfrak{L}_1, \ldots, \mathfrak{L}_k$ and with behavior Υ, if

$$\mathcal{O} : \Omega^u \times \mathbb{N} \to \mathscr{E} \text{ satisfying } {}^n\mathcal{O} \text{ is a subsequence of } {}^{n+1}\mathcal{O} \text{ where } {}^n\mathcal{O} := \mathcal{O}(\omega^u, n) \tag{1.1}$$

for all $n \in \mathbb{N},\ w^u \in \Omega^u$;

$$\mathfrak{L}_\ell : \mathscr{E} \to \mathcal{M}_\ell, \quad \forall \ell \in \{1, \ldots, k\}$$

and

$$\Upsilon : \mathcal{M}_1 \times \cdots \times \mathcal{M}_k \to (\mathcal{D} \to \mathcal{P}(\mathscr{A}))$$

Furthermore, we call

$${}^n\Upsilon : \Omega^u \times \mathbb{N} \to (\mathcal{D} \to \mathcal{P}(\mathscr{A})), \quad {}^n\Upsilon := \Upsilon(\mathfrak{L}_1({}^n\mathcal{O}), \ldots, \mathfrak{L}_k({}^n\mathcal{O}))$$

is the planned behavior of the player at age n.

Note that a common domain $\mathcal{D}$ is not restrictive and might have subdomains that each estimation depends on. As an example, it is standard to have a model with time and state space included in $\mathcal{D}$. Also, tpically $\mathcal{E} \subset \mathbb{A} \times \mathcal{D}$. When there are more than one player, we keep track of them by the index $i \in \mathbb{N}_0 := \{1, 2, \dots\}$, and set $\vec{\varphi} := \prod_{i \in \mathbb{N}_0} \varphi^i$ for any notation in the Definition 1. In this case, typically $\mathcal{E}^i \subset \vec{\mathbb{A}} \times \vec{\mathcal{D}}$ and players are observing each other. As an example, there might be a common observed state space, or players might observe each other's actions.

Now, equip the space $(\mathcal{D} \to \mathcal{P}(\mathcal{A}))$ with a metric d and define the concept of convergence as follows;

Definition 2 *Given a player $(\mathcal{O}, \mathfrak{L}_1, \dots, \mathfrak{L}_k, \Upsilon)$, we say $\Upsilon^* \in (\mathcal{D} \to \mathcal{P}(\mathcal{A}))$ is a (r, δ)-recurrent behavior if*

$$\mathbb{P}^u\Big(\liminf_{n\to\infty} d(\Upsilon^*, {}^n\Upsilon) > r\Big) \leq \delta$$

Also, we say Υ^* is almost surely a recurrent behaviour of the player if $r = \delta = 0$.

We can generalize the definition to more than one player in a straightforward manner. In this case, as they interact with each other, a collectively recurrent behavior forms the basis for an equilibrium. However, to improve the characterization of an equilibrium, one needs to further empose conditions on the estimations too. Throughout this work, we will explore and modify these definitions to settings that are commonly considered in the literature.

Next, let us further assume that the spaces of estimations $\mathcal{M}_1, \dots, \mathcal{M}_k$ are also equipped with metrics $d_1, \dots, d_k$. Then, if Υ is a continuous mapping, it suffices for estimations to have a limit point to have a recurrent behavior.

Lemma 1 *Let $(\mathcal{O}, \mathfrak{L}_1, \dots, \mathfrak{L}_k, \Upsilon)$ be a player in the environment $(\Omega^u, \mathcal{F}^u, \mathbb{P}^u)$, and suppose there exists $(\varphi_1^*, \dots, \varphi_k^*) \in (\mathcal{M}_1, \cdots, \mathcal{M}_k)$ such that*

$$\mathbb{P}^u\Big(\liminf_{n\to\infty} \max_{1\leq\ell\leq k} d_\ell(\mathfrak{L}_\ell({}^n\mathcal{O}), \varphi_\ell^*) = 0\Big) = 1$$

If $\Upsilon : \mathcal{M}_1 \times \cdots \times \mathcal{M}_k \to (\mathcal{D} \to \mathcal{P}(\mathcal{A}))$ is a continuous mapping, then $\Upsilon^* := \Upsilon(\varphi_1^*, \dots, \varphi_k^*)$ is almost surely a recurrent behavior of the player.

In Section 2, we explore the setting of discrete games, introducing domains and estimations typically considered in modeling many of our games. Moreover, we introduce the concept of uncertain equilibrium and demonstrate how to refine it by imposing conditions on the estimations. Later, we provide a two-player game toy example to illustrate the dynamic nature of games even in the simplest settings. In Section 3, we explore the setting of mean-field games with constant estimations, except where the representative player estimates the population strategies. As observations can be generated by relying on symmetries, we introduce a learning algorithm with explicit examples. In Section 4, we adapt the framework from Section 2 to a single-player context and demonstrate that it can be seen as a general version of Markov Decision Processes.

2 Discrete Games

Let $\mathbb{T} = \mathbb{N}$ denote the time indices, define $\mathbb{T}^t_s := \{t, \dots, s\}$ for each $t \leq s \in \mathbb{T}$, and set $\mathbb{T}_t := \mathbb{T}^0_t, \mathbb{T}^t := \mathbb{T}^t_\infty$. Let $\mathbb{S}_t$ be a measurable state space for each $t \in \mathbb{T}$, and define $\mathbb{S} := \bigcup_{t\in\mathbb{T}} \mathbb{S}_t$. For arbitrary set E with Borel σ-algebra $\mathcal{B}(E)$, let $\mathcal{P}(E)$ denote the set of all probability measures on E. We will always consider discrete indexing to avoid discussions on regularities and measurability.

Take $\Omega := \prod_{t\in\mathbb{T}} \mathbb{S}_t$ as the canonical space. Define $X : \mathbb{T} \times \Omega \to \mathbb{S}$ as the canonical process: $X_t : \Omega \to \mathbb{S}_t$, and $X_t(\omega) = \omega_t$ for each $\omega \in \mathbb{X}$ and $t \in \mathbb{T}$. Let $\mathbb{F}^X$ denote the filtration generated by X. We always require any function defined on $\mathbb{T} \times \Omega$ to be Markovian, similar to the canonical process, and denote their parameters as (t, x) where it is understood that $x \in \mathbb{S}_t$. Note that since $\mathbb{S}_t$ is also indexed, this does not impose a restriction. In particular, there is no need to bookkeep paths of the process.

We use $i \in \mathbb{N}_0 := \mathbb{N} \setminus \{0\}$ as the index for players. Naturally, it is common to restrict models to finitely many players, but it is not necessary for our purposes. For any $i \in \mathbb{N}_0$, let $\mathbb{A}^{t,x;i}$ be the action space of player i at $(t, x) \in \mathbb{T} \times \mathbb{S}$. Introduce

$$\vec{\mathbb{A}}^{t,x} := \prod_{i\in\mathbb{N}_0} \mathbb{A}^{t,x;i}, \qquad \vec{\mathbb{A}} := \bigcup_{(t,x)\in\mathbb{T}\times\mathbb{S}} \vec{\mathbb{A}}^{t,x}, \qquad \mathbb{A}^i := \bigcup_{(t,x)\in\mathbb{T}\times\mathbb{S}} \mathbb{A}^{t,x;i}.$$

Let us also introduce the space of controls;

$$\mathcal{A}^i := \left\{\alpha : \mathbb{T} \times \Omega \to \mathbb{A}^i \ : \ \alpha(t, x) \in \mathbb{A}^{t,x;i} \ \ \forall (t, x) \in \mathbb{T} \times \mathbb{S}_t\right\}, \quad \forall i \in \mathbb{N}_0$$

and set $\vec{\mathcal{A}} := \prod_{i\in\mathbb{N}_0} \mathcal{A}^i$.

As for the learning parameters, we will now begin to introduce horizon, transitions between states, transition costs, state values, potential behaviors of other players, optimal controls and expectations of the players. Our choices are inherently limited as a player might be arbitrarily complicated, and includes some modeling choices, but general enough to cover a diverse set of examples and to demonstrate the concept of uncertain equilibrium. It is important to note that our primary goal is not to model the learning of these parameters, as each parameter could constitute its own line of research. For simplicity, we will temporarily disregard the index i and focus solely on the perspective of a single player.

First of all, players need to have a *horizon* $\hat{T}$. We assume that, as always the case in reality, players cannot predict the future indefinitely with reasonable accuracy. In other words, as the horizon of prediction increases, the distribution of the state process contains progressively less useful information, eventually rendering it useless. Even under the infinite horizon settings, typically one introduces a discounting factor, essentially approximating a finite horizon. Thus, let

$$\mathcal{M}_T := \left\{\hat{T} : \mathbb{T} \times \Omega \to \mathbb{T}\right\} \tag{2.1}$$

be the space of all such functions, where the corresponding learning algorithm will take values in. Notice that we allowed the horizon to depend on the state, since the player might be able to project further in well-trained states. More importantly, rather than a fixed time, one can consider a stopping time $\hat{T}^{(t,x)}(s,y) : (\mathbb{T} \times \Omega)^2 \to \mathbb{T}$, where $\hat{T}^{(t,x)}$ is a stopping time for the future estimated process in (2.4).

Next, the player might have an estimate of the transition probabilities;

$$\begin{aligned} &\hat{p} : \mathbb{T} \times \Omega \times \vec{\mathbb{A}} \times \mathbb{S} \to \mathbb{R}^+, \quad \text{where} \\ &\hat{p}(t,x,\vec{a};\cdot) \ \text{ is a probability measure on } \ \mathbb{S}_{t+1}, \ \text{ for all } t \in \mathbb{T}, x \in \mathbb{S}_t, \ \text{and } \vec{a} \in \vec{\mathbb{A}}^{t,x} \end{aligned} \tag{2.2}$$

Similarly, introduce $\mathcal{M}_p$ as the space of all such mappings in (2.2). Note that Large Language Models excel in the task of constructing $\hat{p}$ and solve the universal problem of predicting the next input in the specific context of language.

It is crucial that players learn about other players' behavior. To fully understand any complex game, we cannot overlook this fact. Knowledge of opponents' strategies intrinsically alters the observed events within the game. Even a player's value depends on it, as different opponents might tend to employ varying strategies. Consequently, the value associated with a strategy cannot disregard the opponents' reactions. Thus, we assume that a player learns potential controls of others based on their own control;

$$\hat{\Gamma}^i : \mathbb{T} \times \mathcal{A}^i \to \mathcal{P}(\vec{\mathcal{A}}) \quad \text{and set } \ \hat{\Gamma}^i_{t,\alpha}(d\vec{\alpha}) := \hat{\Gamma}^i_t(\alpha; d\vec{\alpha}) := \hat{\Gamma}^i(t,\alpha)(d\vec{\alpha}) \tag{2.3}$$

Denote $\mathcal{M}_\Gamma$ as the space of all such mappings. We remark two points for (2.3):

(i) Here, we assume that players model the others potential controls depending on their own control. However, one might model that this depends on the path of states of the players, or in fact, any other observables are legitimate as long as the cost (2.6) is well-defined. Our choice here is to illustrate the optimization in a visible manner.

(ii) It is crucial for players to learn reliable $\hat{\Gamma}$. To not only compete with but also cooperate with other players, they must generate reliable estimates of the actions of others. We will show in the two-player game discussed in Section 2.1 that because the costs to the players depend on each other's states, omitting this aspect from the player model won't accurately capture the observed dynamics.

Given $\hat{p}$ as in (2.2), an initial $(t,x) \in (\mathbb{T}, \mathbb{S}_t)$, and $\vec{\alpha} \in \vec{\mathcal{A}}$, player induces a distribution $\mathbb{P}^{t,x,\vec{\alpha}} := \mathbb{P}^{\hat{p};t,x,\vec{\alpha}}$ for the canonical process as usual; for all $t \le s$ and $(\tilde{x}, y) \in (\mathbb{S}_s, \mathbb{S}_{s+1})$,

$$\mathbb{P}^{t,x,\vec{\alpha}}(X_{s+1} = y | X_s = \tilde{x}) = \hat{p}(s, \tilde{x}, \vec{\alpha}(s,\tilde{x}); y), \ \text{ and } \ \mathbb{P}^{t,x,\vec{\alpha}}(X_t = x) = 1 \tag{2.4}$$

Note that relaxed controls further integrate over the distribution of controls to define (2.4). We instead integrate the value below.

A crucial notion to introduce is the value of a player. We observe that value is fundamentally a formalism used to determine an optimal action. We do not regard it as something tangible that a player necessarily obtains, which might only be the case in limited situations. However, even in these situations, value is typically considered an expectation rather than a definitive quantity. Now, we introduce two concepts: transition costs and state values, which are akin to running costs and terminal costs in the literature[1]:

$$\hat{F} : \hat{\Omega} \times \mathbb{T} \times \Omega \times \vec{\mathbb{A}} \to \mathbb{R}, \qquad \hat{\phi} : \hat{\Omega} \times \mathbb{T} \times \Omega \to \mathbb{R}, \tag{2.5}$$

and let $\mathcal{M}_F, \mathcal{M}_\phi$ denote the sets of mappings as in (2.5). An important difference is that the player models these as random variables on some probability space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$. In particular cases, it might be useful to characterize the measure space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$, however, once can also fix the sufficiently large probability space $([0,1], \mathcal{B}([0,1]), \mathbf{m})$ and concentrate on the random variables. We remark that state value $\hat{\phi}$, in particular, induces an ordering on states. Moreover, reaching a particular state by different intermediate paths, or different set of actions might have varying costs, which is aimed to be captured by the transition cost $\hat{F}$.

Now, given $(\hat{T}, \hat{p}, \hat{\Gamma}, \hat{F}, \hat{\phi})$, the value of player becomes

$$\begin{aligned} &J(t,x;\alpha) := \int_{\vec{A}} J(t,x;\vec{\alpha}) \hat{\Gamma}_t(\alpha; d\vec{\alpha}) \;\text{ where} \\ &J(t,x;\vec{\alpha}) := \mathbb{E}^{t,x,\vec{\alpha}}\Big[\hat{\phi}(t+\hat{T}, X_{t+\hat{T}}) + \sum_{s=t}^{t+\hat{T}-1} \hat{F}(s, X_s, \vec{\alpha}(s, X_s))\Big], \quad \mathbb{E}^{t,x,\vec{\alpha}} := \mathbb{E}^{\mathbb{P}^{t,x,\vec{\alpha}}} \end{aligned} \tag{2.6}$$

which is a random variable on $\hat{\Omega}$. Set $\mathcal{M}^i_J$ as the space of all such functions $(\hat{\Omega} \times \mathbb{T} \times \Omega \times \mathcal{A}^i \to \mathbb{R})$.

Let us recall the game of chess, which serves as an excellent example to keep in mind throughout this work. In chess, $\hat{p}$ yields deterministic transitions. However, a player does not know what actions the opponent will take within $\{t, \ldots, t+\hat{T}\}$, and beyond that, it is unclear what the transition costs of actions or the value of being in a particular state at $t+\hat{T}$ might be. These are all crucial components for a player to learn. Notably, the heuristic values assigned to pieces are designed to guide players in learning $\hat{F}$ and $\hat{\phi}$. While simplistic, these heuristics serve as an initial guide. Moreover, as we have mentioned, knowledge about the opponent can alter the values of strategies, which is captured in (2.6). Let us also emphasize that the player's horizon may depend significantly on the current state. Towards the endgame, for instance, a well-trained chess player might be able to estimate many steps ahead, whereas this ability may be considerably more limited during the middle stages of the game.

[1] We use cost and value interchangeably. In the case of scalar objectives as in this work, distinction is more pronounced. However, for multi-objective frameworks, there is typically no binary choice, but rather a continuum of choices.

We remark that it is obviously equivalent to define

$$J(t,x;\vec{\alpha}) := \mathbb{E}^{t,x,\vec{\alpha}}\Big[\hat{\phi}(t+\hat{T}, X_{t+\hat{T}}) - \hat{\phi}(t,x) + \sum_{s=t}^{t+\hat{T}-1} \hat{F}(s, X_s, \vec{\alpha}(s, X_s))\Big]$$

which is more intuitive and useful in computations, but we won't keep track of it. Moreover, one can further extend the space of parameters. Simplest example might be to write $\hat{F}(t,s,x,a)$. We are trying to point out various modeling choices that we made to cover enough cases to promote our framework; however, it is necessary to adjust these appropriately based on the problem under consideration.

As the player faces the optimization problem (2.6), it is not always feasible to solve for the optimal control. When $\hat{T} = 1$, the problem might be relatively simple, allowing for straightforward searches for ϵ-optimal actions. However, for longer horizons, the space of potential controls becomes excessively large, complicating the search for optimal solutions. To formalize this, let us first define

$$\alpha =^{t,x;i} \tilde{\alpha} \quad \text{if} \quad \alpha(s,y) = \tilde{\alpha}(s,y) \quad \forall s \in \{t, \dots, t + \hat{T}^i(t,x) - 1\},\ y \in \mathbb{S}_s$$

Under this equivalency relation, we introduce the quotient space

$$\mathcal{A}^{t,x;i} := \mathcal{A}^i / =^{t,x;i}$$

And then, to incorporate the potential difficulty and uncertainty in identifying the optimal control, we introduce the next learning parameter;

$$\begin{aligned} &\hat{\pi} : \hat{\Omega} \times \mathbb{T} \times \Omega \to \mathcal{P}(\mathcal{A}^i) \qquad \text{where,} \\ &\hat{\pi}(\hat{\omega}, t, x)(d\alpha) = \hat{\pi}(\hat{\omega}, t, x)(d\tilde{\alpha}) \ \text{ whenever } \ \alpha =^{t,x;i} \tilde{\alpha} \end{aligned} \tag{2.7}$$

Note that, in general, the equality should be understood in terms of two subsets of controls having equal probabilities, when they are equal once extended with the relation $=^{t,x;i}$. Also, we didn't suppress $\hat{\omega}$ as it will be integrated. Here, at $(\hat{\omega}, t, x)$, $\hat{\pi}$ approximates the potential optimal controls for $J(\hat{\omega}, t, x, \cdot)$, which will be dictated by the equilibrium condition below.

Now, the crucial observation is that, even when optimal control can be solved exactly, uncertainty over the value will naturally induce a probability distribution over controls. That is, given $(\hat{T}, \hat{p}, \hat{\Gamma}, \hat{F}, \hat{\phi}, \hat{\pi})$, we define $\Upsilon^{t,x;i} \in \mathcal{P}(\mathcal{A}^{t,x;i})$ as

$$\Upsilon^{t,x;i}(d\alpha) := \int_{\hat{\Omega}} \hat{\pi}(\hat{\omega}, t, x)(d\alpha) \hat{\mathbb{P}}(d\hat{\omega}) \tag{2.8}$$

We have introduced distributions over controls, which then further induces distribution over actions:

$$\gamma^{t,x;i}(da) := \Upsilon^{t,x;i}\big(\alpha : \ \alpha(t,x) = da\big) \in \mathcal{P}(\mathbb{A}^{t,x;i}) \tag{2.9}$$

Although apriori $\Upsilon^{t,x;i}$ appears to be solely induced by $\hat{\pi}$, $\hat{\pi}$ itself is a function of the value J^i, and hence $\Upsilon^{t,x;i}$ is a function from $\mathcal{M}_T \times \mathcal{M}_p \times \mathcal{M}_\Gamma \times \mathcal{M}_F \times \mathcal{M}_\phi \times \mathcal{M}_\pi$. In the two-player example discussed in section 2.1, $\hat{\pi}$ solves the optimization by brute force, yielding a deterministic action for each scenario in $\hat{\Omega}$. In the control problem described in section 4, $\hat{\pi}$ constructs the distribution over strategies relying again on the associated values. Let us note that we induce a distribution over controls because it is a more familiar and convenient choice; however, by considering sequences of future states, one can easily induce a distribution over sequences of actions. Moreover, this behavior, represented by Υ, is well-suited for the learning or playing phase. One might want to modify the behavior of the player to yield the most likely control during the competition phase.

It is important to motivate the role of randomness in value, which then induces a probability distribution over actions in (2.8). Recall that in sufficiently complex settings, such as chess, values are inherently unknown and must be learned through significant effort. That is, the randomness of the value models what is unknown to the player. The key role of randomness in value is to allow players to explore systematically. If the player is not satisfied with the current estimates, it is natural to assign greater probabilities to unexplored controls for having higher values. This approach naturally leads to the search for controls with more satisfactory results. We will demonstrate a toy version for the two-player game in Section 2.1. Consider a daily life scenario: suppose you have a favorite dish, and your friend suggests a new one you have never tried. Since you expect it is not better, if your model only considers expectations, you won't explore. However, if you model the value as random, you might try the new dish. This approach aligns with the common intuition that a better understanding of values should lead to less uncertain strategies. Once you try the new dish, you know its value to you more accurately, and that experience might deter further exploration. Again, the design of a player is a crucial aspect of understanding games.

Now that we have introduced the spaces of estimations and behaviors, let us also recall the observations. These observations might be coming from real world experience. Or in the mean-field regime, players can generate observations by assuming every other player is exactly the same. To distinguish the index of observations and learning from time, we reserve $n \in \mathbb{N}$ written on the left superscripts. Let $\mathcal{E}^i$ denote the space of observables for player i, and set $\mathscr{E}^i$ as the finite sequences of $\mathcal{E}^i$. Similar to previous notations, set $\vec{\mathcal{E}} := \prod_{i \in \mathbb{N}_0} \mathcal{E}^i$ and $\vec{\mathscr{E}} := \prod_{i \in \mathbb{N}_0} \mathscr{E}^i$. Now, observations are given as

$$\mathcal{O}^i : \Omega^u \times \mathbb{N} \to \mathscr{E}^i \text{ where } {}^n\mathcal{O}^i \text{ is a subsequence of } {}^{n+1}\mathcal{O}^i, \ \forall i \in \mathbb{N}_0,\ n \in \mathbb{N},\ w^u \in \Omega^u \quad (2.10)$$

and set $\vec{\mathcal{O}} = (\mathcal{O}^1, \mathcal{O}^2, \dots)$. Recall that we assume there exists a universal probability distribution $\mathbb{P}^u$ on some arbitrary Ω^u, which might be induced by real dynamics of the universe that players are in and distributions players are drawing from. We say that ${}^n\mathcal{O}^i$ characterizes all the observations of player i up to age n.

Now, we want to acknowledge the existence of a learning algorithm. We say a collection of functions $\mathfrak{L}^i_\varphi$ for $\varphi \in \{T, p, \Gamma, F, \phi, \pi\}$ is the learning algorithm of player i. Recall that $\mathcal{M}_T, \mathcal{M}_p, \mathcal{M}_\Gamma, \mathcal{M}_F, \mathcal{M}_\phi, \mathcal{M}_\pi$ are the spaces of estimations as in (2.1), (2.2), (2.3), (2.5), and (2.7) respectively. Then, we let

$$\mathfrak{L}^i_\varphi : \mathcal{E}^i \to \mathcal{M}_\varphi, \quad \forall \varphi \in \{T, p, \Gamma, F, \phi, \pi\} \tag{2.11}$$

Moreover, for any function $\mathfrak{L}^i_\varphi$ on $\mathcal{E}^i$, we set *estimations* of the player as

$${}^n\hat{\varphi}^i := {}^n\mathfrak{L}^i_\varphi := \mathfrak{L}^i_\varphi({}^n\mathcal{O}^i), \quad \forall \varphi \in \{T, p, \Gamma, F, \phi, \pi\}$$

and call ${}^0\hat{\varphi}^i$ as the prior of player i. Note that, $({}^n\hat{T}^i, {}^n\hat{p}^i, {}^n\hat{\Gamma}^i, {}^n\hat{F}^i, {}^n\hat{\phi}^i)$ defines ${}^nJ^i_{\mathfrak{L}}(t,x;\alpha)$ as in (2.6), which then together with ${}^n\hat{\pi}^i$ induces $\Upsilon^{t,x;i}_{\mathfrak{L}}$ on ${}^n\mathcal{A}^{t,x;i}$ as in (2.8) and ${}^n\gamma^{t,x;i}_{\mathfrak{L}}$ on $\mathbb{A}^{t,x;i}$ as in (2.9).[2]

We say *all players are behaving rationally*, if at each $(t,x) \in \mathbb{T} \times \mathbb{S}$, they draw an action from the distribution ${}^n\vec{\gamma}^{t,x}_{\mathfrak{L}} := ({}^n\gamma^{t,x;1}_{\mathfrak{L}}, {}^n\gamma^{t,x;2}_{\mathfrak{L}}, \dots)$, or draw from the distribution ${}^n\Upsilon^{t,x}_{\mathfrak{L}} := ({}^n\Upsilon^{t,x;1}_{\mathfrak{L}}, {}^n\Upsilon^{t,x;2}_{\mathfrak{L}}, \dots)$ to decide on strategies for $\mathbb{T}^t_{t+\hat{T}}$. We always assume that players are learning under rational behaviors. In a two-player game, for example, if one player behaves nonsensically, expecting any type of convergent behavior becomes meaningless. We emphasize that this assumption of rationality is not a constraint but rather means that we model players within the current framework of this work.

For notational convenience, instead of explicitly denoting $\{T, p, \Gamma, F, \phi, \pi\}$, we keep using φ to indicate any one or all of the estimations, and denote $\vec{\varphi} := (\varphi^1, \varphi^2, \cdots)$.

Definition 3 (Uncertain Equilibrium of Discrete Games) *We say $\vec{\varphi} \in \mathcal{M}^{\mathbb{N}_0}_\varphi$ is an (ε, r, δ)-uncertain equilibrium at $(t,x) \in \mathbb{T} \times \mathbb{S}_t$ under the learning algorithm $\vec{\mathfrak{L}}_\varphi$ if,*

(i) $\vec{\varphi}$ are the priors of players,

(ii)

$$\int_{\hat{\Omega}} \int_{\mathcal{A}^i} \Big(\sup_{\tilde{\alpha} \in \mathcal{A}^i} {}^nJ^i(\hat{\omega}, t, x, \tilde{\alpha}) - {}^nJ^i(\hat{\omega}, t, x, \alpha) \Big) {}^n\hat{\pi}^i(\hat{\omega}, t, x)(d\alpha) \hat{\mathbb{P}}(d\hat{\omega}) \le \varepsilon, \quad \forall i \in \mathbb{N}_0, n \in \mathbb{N}$$

(iii)

$$\mathbb{P}^u\Big(\liminf_{n\to\infty} \sup_{i \in \mathbb{N}_0} d^{t,x;i}({}^0\Upsilon^{t,x;i}_{\mathfrak{L}}, {}^n\Upsilon^{t,x;i}_{\mathfrak{L}}) > r \Big) \le \delta$$

where $d^{t,x;i}$ is the metric player i equips the space $\mathcal{P}(\mathcal{A}^i)$ under the equivalence $=^{t,x;i}$.

[2] We may drop $\mathfrak{L}$ from subscripts if it is clear from the context.

We say $\vec{\varphi} \in \mathcal{M}_{\varphi}^{\mathbf{N}_0}$ is an (ε, r, δ)-uncertain equilibrium if it is an (ε, r, δ)-uncertain equilibrium for all $(t, x) \in \mathbb{T} \times \mathbb{S}_t$.

We remark that condition (iii) aligns with the abstract notion of convergence defined in the Introduction. We have presented it by relying on the priors of players, as this approach is more intuitive and aligns better with the established notations. In essence, we are imposing a further condition (ii) on estimations to achieve a more favorable concept of equilibrium. Later in this section, we will introduce an additional learning parameter and discuss how to incorporate it into this definition.

As a first example, suppose there exists finitely many models $\{\vec{\varphi}_m\}_{m=1}^{M}$ for which $\vec{\mathfrak{L}}_{\varphi}$ can take values in. Furthermore,

$$\mathbb{P}^u\big({}^{n+1}\vec{\mathfrak{L}}_{\varphi} = \vec{\varphi}_m | \, {}^{n}\vec{\mathfrak{L}}_{\varphi} = \vec{\varphi}_{\ell}\big) > 0, \quad \forall \ell, m \in \{1, \dots, M\}, \; n \in \mathbb{N}, \varphi \in \{T, p, \Gamma, F, \phi, \pi\}$$

then, all of the models satisfy (iii) in the definition of uncertain equilibrium. As long as players are able to solve their own optimization problem, all the models are uncertain equilibrium. Of course, in reality it is unlikely to have finitely many potential models. Basic games like rock-paper-scissors can be modeled in this way. Here, only relevant model is about what the other player will do, that is Γ^1, Γ^2. Then by construction, we can simplify to only 3 different models assigning certain actions is enough. Then, depending on the ongoing observations, players will continuously update their estimations on Γ and act accordingly.[3]

Prior to discussing the definition, let us revisit the example of chess again. Consider a well trained chess player, and suppose the game is nearing its end. At this late stage, there are potentially many configurations where the subsequent moves are certain. That is, a particular action has an induced probability of 1, and it remains unchanged as the player continues to learn. We then say that the player is in equilibrium at those particular end game configurations. Similarly, at the very beginning of the game, the player might have a distribution over different openings. Although we will not observe the same opening in each game, for a well-trained player, the distribution over these openings may evolve only over long time scales. We also recognize this as an equilibrium. On the other hand, there may be many other configurations where learning ever continues.

To briefly elaborate on how players solve their own optimization problems, this is partly achieved by repeating past observations with evolving estimations. As a player learns, i.e. as n increases, and recalls past observations, the exploration of other potential scenarios under ${}^{n}\hat{\varphi}^{i}$ aims to capture the term $\sup_{\alpha \in \mathcal{A}^i} {}^{n}J^{i}(\hat{\omega}, t, x, \alpha)$. During this revaluation and exploration process, new strategies may be discovered, or a new assessment of value might lead to changes in ${}^{n}\hat{\pi}^{i}$. Condition (ii) in the definition of uncertain equilibrium implies that the player has explored potential strategies and is capable of generating the best ones under various scenarios of $\hat{\Omega}$ as learning continues.

[3] This may not be a natural way to model, but rather to provide a straightforward example.

On the other hand, the values of the player are driven externally. There is no universal concept of what holds higher value; rather, these concepts are shaped by factors such as needs, interactions, and self-evaluations, as evidenced by the highly diverse values found across individuals and societies. We would like to emphasize that the equilibrium can be understood specifically for each $(t, x) \in \mathbb{T} \times \mathbb{S}_t$. With this context in mind, as the player considers a past state that could potentially recur, condition (iii) in the definition of uncertain equilibrium implies that the induced distribution over strategies also recurs as learning progresses.

The choice of liminf instead of limsup is important. Let us consider two scenarios:

(i) Players might remain in a particular equilibrium for nearly all of the time as learning progresses, but they may occasionally explore other actions. Such behavior is observable even in the simplest settings, as we will demonstrate later. We still recognize this as an equilibrium. The use of liminf implies that the players are within an r-distance of this particular equilibrium infinitely often.

(ii) There might be two or more equilibria between which players switch. We recognize each one as an equilibrium, as players will find themselves in each particular equilibrium infinitely often. If we were to require limsup instead, it would imply that players are stuck in a particular equilibrium indefinitely, which is quite hard to expect in reality.

Let us briefly explain the role of (ε, r, δ), which we took uniform over players for simplicity. Firstly, ε characterizes how effectively players can solve their targeted optimization problem. Next, r measures how closely the distribution of strategies are approaches the equilibrium infinitely often. Lastly, δ reflects the likelihood that an equilibrium will be observed. Recall that $\mathbb{P}^u$ is associated with the universe in which the players exist; this may depend on the real underlying dynamics, random choices generated by each player etc.

Let us emphasize again that the concept of equilibrium is characterized by the recurrence of the induced strategy distributions of players. In fact, we remark that ${}^n\Upsilon^{t,x;i}_{\mathfrak{L}}$ can be viewed as a function $\mathbb{N} \times \Omega^u \to \mathcal{P}(\mathcal{A}^i)$, thus a discrete time stochastic process taking values in the space of probability distributions over $\mathcal{A}^i$. Therefore, the uncertain equilibrium is precisely a recurrent point if $r, \delta = 0$. The primary interest lies in understanding the evolution of ${}^n\Upsilon^{t,x;i}_{\mathfrak{L}}$ under a specific learning algorithm, which characterizes the observed behaviors. This offers an alternative perspective for relaxed controls. Indeed, we aim to consider random strategies because the values are inherently uncertain. This is again familiar as better comprehension of the values allows for the construction of more certain strategies.

Now, we will clarify the similarities and differences with the concept of correlated equilibrium. To do so, let us look at the optimality condition in the definition of uncertain equilibrium 3 as follows, simplifying notation by omitting n and (t, x):

$$\begin{aligned}&\int_{\hat{\Omega}}\int_{\mathcal{A}^i} J^i(\hat{\omega},\alpha)\hat{\pi}^i(\hat{\omega})(d\alpha)\hat{\mathbb{P}}(d\hat{\omega})\\ &=\int_{\hat{\Omega}}\int_{\mathcal{A}^i}\int_{\vec{\mathcal{A}}} J^i(\hat{\omega},\vec{\alpha})\hat{\Gamma}^i(\alpha;d\vec{\alpha})\hat{\pi}^i(\hat{\omega})(d\alpha)\hat{\mathbb{P}}(d\hat{\omega})\end{aligned}$$

and one notices that

$$\int_{\hat{\Omega}}\int_{\mathcal{A}^i} \hat{\Gamma}^i(\alpha;d\vec{\alpha})\hat{\pi}^i(\hat{\omega})(d\alpha)\hat{\mathbb{P}}(d\hat{\omega}) \in \mathcal{P}(\vec{\mathcal{A}}) \tag{2.12}$$

Note that the i-th marginal of (2.12) is Υ^i, the induced distribution of player i. To clarify the connections, consider any $\rho \in \mathcal{P}(\vec{\mathcal{A}})$. By disintegrating ρ onto the i-th component as $\rho(d\vec{\alpha}) = \rho^{-i}(d\vec{\alpha}|\alpha^i)\rho^i(d\alpha^i)$, we have that ρ^{-i} corresponds to $\hat{\Gamma}^i$[4] and ρ^i corresponds to $\hat{\pi}^i(\hat{\omega})(d\alpha)\hat{\mathbb{P}}(\hat{\omega})$. Roughly speaking, equilibrium conditions are (with simplified notations),

Nash-type Equilibrium: $$\int_{\mathcal{A}^i}\int_{\vec{\mathcal{A}}}\sup_{\tilde{\alpha}^i} J^i(\tilde{\alpha}^i,\vec{\alpha}^{-i})\rho^{-i}(d\vec{\alpha}|\alpha^i)\rho^i(d\alpha^i)$$

Correlated Equilibrium: $$\int_{\mathcal{A}^i}\sup_{\tilde{\alpha}^i}\int_{\vec{\mathcal{A}}} J^i(\tilde{\alpha}^i,\vec{\alpha}^{-i})\rho^{-i}(d\vec{\alpha}|\alpha^i)\rho^i(d\alpha^i)$$

Uncertain Equilibrium: $$\int_{\hat{\Omega}}\sup_{\tilde{\alpha}^i}\int_{\vec{\mathcal{A}}} J^i(\hat{\omega},\tilde{\alpha}^i,\vec{\alpha}^{-i})\rho^{-i}(d\vec{\alpha}|\tilde{\alpha}^i)\hat{\mathbb{P}}(d\hat{\omega})$$

Coarse Correlated Equilibrium: $$\sup_{\tilde{\alpha}^i}\int_{\mathcal{A}^i}\int_{\vec{\mathcal{A}}} J^i(\tilde{\alpha}^i,\vec{\alpha}^{-i})\rho^{-i}(d\vec{\alpha}|\alpha^i)\rho^i(d\alpha^i)$$

where the supremum over $\tilde{\alpha}^i$, for the Nash-type equilibrium, depends on $\vec{\alpha}^{-i}$; for the correlated equilibrium, it depends on α^i; for the uncertain equilibrium, it depends on $\hat{\omega}$; and for the coarse correlated equilibrium, it is independent of the controls. Note that we have changed $\hat{\pi}$ to $\sup_{\tilde{\alpha}^i}$ due to the optimality condition, and $\hat{\mathbb{P}}$ plays the same role as ρ^i. However, the key difference is that the correlated equilibrium, similar to the Nash equilibrium, considers deviations without affecting other players, whereas in uncertain equilibrium, changing the strategy would affect others.

Recall that if regret is defined similarly to the correlated equilibrium and is assumed to be sublinear, then by definition, the observed time-averaged joint distribution over actions converges to a correlated equilibrium. Considering the estimated distributions over strategies in (2.12), one might expect that all players will eventually induce the same distribution in symmetric situations. However, in general, there is no reason to assume that a single distribution will characterize every player's considerations.

The concept of Nash equilibrium focuses solely on controls, according to their associated scalar values, inherently excluding the intrinsic structure of a player. In situations where a central planner

[4] One can extend ρ^{-i} to the full $\vec{\mathcal{A}}$ by the Dirac distribution on α^i for the i-th marginal, which is also the case for $\hat{\Gamma}^i$.

announces policies for individual agents, such as environmental regulations, traffic management, public health initiatives, with the knowledge that every individual will act according to their own assesment of value, Nash equilibrium is definitely the right consideration. The central planner needs to model agent's individual values to construct stable policies. In such cases, it is not meaningful to model each and every agent in detail of their learning algorithms. We also remark that, the Nash equilibrium requires players to have exact knowledge of the strategies of other players. In a pure equilibrium sense, this instability is significant enough that an equilibrium does not exist even for simplest games like rock-paper-scissors. To address this, one typically adopts relaxed controls, which is essential but still omits details about the underlying player.

Let us highlight the importance of incorporating additional learnable parameters into our framework. This could include encoding raw observations, establishing communication protocols, and many other spaces of estimations to design more sophisticated player. One such crucial parameter is the players' expectations regarding the best achievable outcomes, which define a notion of regret for the player and alter the characteristics of exploration, as demonstrated in a simplified form in Section 2.1. Consider functions of the form

$$\hat{B} : \mathbb{T} \times \Omega \to \mathbb{R} \tag{2.13}$$

and all the related definitions similar to other learning parameters. Then, we can introduce;

$${}^{n}\kappa^{i}(t,x) := \hat{\mathbb{P}}\left(\int_{\mathcal{A}^i} {}^{n}J^{i}(\hat{\omega},t,x,\alpha)^{n}\hat{\pi}^{i}(\hat{\omega},t,x)(d\alpha) > {}^{n}\hat{B}^{i}(t,x) \right)$$

To relate this quantity to familiar concepts with which we all can relate, we say that at the state $(t,x) \in \mathbb{T} \times \Omega$, the player i is currently desperate if ${}^{n}\kappa^{i}(t,x) = 0$, and euphoric if ${}^{n}\kappa^{i}(t,x) = 1$. If the player is desperate, as the learning progresses, either ${}^{n}\hat{B}^{i}$ will decrease, leading the player, in some sense, to accept the situation; or ${}^{n}J^{i}$ will assign higher values to underexplored strategies, encouraging the player to explore them. Let us note that it is not necessarily better for the learning algorithm to adjust ${}^{n}J^{i}$, since ${}^{n}\hat{B}^{i}$ might be unrealistically high. One can describe the player's current situation in more detail by using verbal subcategories like

Desperate—Discouraged—Doubtful—Cautious—Hopeful—Determined—Confident—Optimistic—Euphoric

which can be set as partitions of κ values. Beyond better characterization of a player, we can incorporate this into the definition of equilibrium 3 by requiring

(ii')

$${}^{n}\kappa^{i}(t,x) > \kappa, \quad \forall i \in \mathbb{N}_0, n \in \mathbb{N}$$

and denoting it as $(\kappa, \varepsilon, r, \delta)$-uncertain equilibrium. That is, we are now searching for an equilibrium where each player is, let's say, at least confident.

As well as the design of the player, choices for the concept of equilibrium are also diverse. For example, to recover the concept of Nash equilibrium, we can further assume

(ii")

$$\operatorname{supp}\left({}^n\hat{\Gamma}^i_{t,x,\alpha}\right) \subset \operatorname{supp}\left({}^n\Upsilon^{t,x,1} \times {}^n\Upsilon^{t,x,2} \times \cdots\right), \quad \forall i \in \mathbb{N}_0, n \in \mathbb{N}, \alpha \in \operatorname{supp}{}^n\Upsilon^{t,x,i}$$

where we extended $\hat{\Gamma}$ to depend on x naturally, and × denotes the product measure. That is, players' estimates are indeed within the support of the induced distributions of all the players. If every estimate is predetermined and no randomness is involved, then (ii") means that players are able to observe each other's future strategies. Thus, together with (ii) in Definition 3, we recovered the standard Nash equilibrium.

Next, time-consistency, or Dynamic Programming Principle, can be naturally introduced for any learning parameter. The most important and familiar one is for the value function, and it also relates $\hat{\phi}$ to the concept of the standard value function in the literature. We say that the estimate $(\hat{T}, \hat{p}, \hat{\Gamma}, \hat{F}, \hat{\phi}, \hat{\pi})$ yields a time consistent value almost surely, if for any $(t, x) \in \mathbb{T} \times \mathbb{S}_t$ and $t \le T_0 \le t + \hat{T}(t, x)$, it holds

$$\int_{\mathcal{A}^i} J(T_0; \hat{\omega}, t, x, \alpha)\hat{\pi}(\hat{\omega}, t, x)(d\alpha) = \int_{\mathcal{A}^i} J(\hat{\omega}, t, x, \alpha)\hat{\pi}(\hat{\omega}, t, x)(d\alpha) \quad d\hat{\mathbb{P}} - a.s. \tag{2.14}$$

where $J(T_0; \hat{\omega}, t, x, \alpha)$ is defined exactly as in (2.6), where $\hat{T}$ is replaced by T_0. Notice that, setting $T_0 = t$ and in case $\hat{\pi}$ yields the optimal control per $\hat{\omega} \in \hat{\Omega}$, (2.14) becomes

$$\hat{\phi}(t, x) = \sup_{\alpha} J(t, x, \alpha) = \sup_{\alpha} \int_{\vec{\mathcal{A}}} \mathbb{E}^{t,x,\vec{\alpha}}\Big[\hat{\phi}(t + \hat{T}, X_{t+\hat{T}}) + \sum_{s=t}^{t+\hat{T}-1} \hat{F}(s, X_s, \vec{\alpha}(s, X_s))\Big]\hat{\Gamma}_t(\alpha; d\vec{\alpha})$$

which closely resembles the standard time-consistency, or Dynamic Programming Principle, for the standard value function. Similarly, for example, we say that the estimate $(\hat{T}, \hat{p}, \hat{\Gamma}, \hat{F}, \hat{\phi}, \hat{\pi})$ yields a time-consistent distribution over controls almost surely at $(t, x) \in \mathbb{T} \times \mathbb{S}_t$, if for any $t \le T_0 \le t + \hat{T}(t, x) - 1$, and $x_{T_0} \in \mathbb{S}_{T_0}$, $\hat{\pi}(\hat{\omega}, t, x)$ induces the same distribution as $\hat{\pi}(\hat{\omega}, T_0, x_{T_0})$ on the space $\mathcal{A}^{\hat{T}(t,x);T_0,x_{T_0},i}$. Here, the quotient space is defined similarly, with the relation terminating at $t + \hat{T}(t, x) - 1$ instead of $t + \hat{T}(T_0, x_{T_0}) - 1$.

2.1 Two player game example

In this section, we present a simple, repeatedly played two-player example. We demonstrate that even with a fixed one-step horizon, players can exhibit sophisticated dynamics. In this setting, both players learn transition costs $\hat{F}$ and the actions of their opponents $\hat{\Gamma}$, all within a fixed horizon $T = 1$. Our primary argument is that formulating controls as an equilibrium does not adequately capture the dynamic strategies continually employed by the players. To address this shortcoming, we construct learning algorithms that capture these dynamic strategies. This example serves to illustrate why a more general framework is necessary for effectively modeling games, and it heuristically explores how the concept of equilibrium is inherently influenced by the learning process and the opponent itself.

Let us point out that, with a one-step horizon, a unique Nash equilibrium exists that the first player is unwilling to play. Of course, by virtue of Folk's theorem, any feasible outcome can be sustained in an infinitely repeated game. However, this result explicitly relies on the assumption that players are completely certain about their opponents' future actions over an indefinite horizon. This strong assumption allows for almost any feasible value to be supported as an equilibrium, leaving the question of which outcome will be observed without a clear answer. Instead, we emphasize once again that the core element in games is the learning algorithms employed by the players. These algorithms naturally govern the (random) evolution of probability distributions over current and future actions, which in turn is sufficient to understand the evolution of the game, provided that players act rationally. Here, ”rational” does not necessarily mean ”clever” since players might easily adopt suboptimal learning algorithms. The design of robust learning algorithms is an area of significant and growing interest.

Consider a fixed state and action space as $\mathbb{S} = \{0, 1\}$ and $\mathbb{A} = [0, 1]$. Suppose players are not learning the horizon and transition probabilities, that is, $\mathfrak{L}^i_T, \mathfrak{L}^i_p$ are constants yielding $\hat{T}^i = 1$ and $p^i(t, \vec{x}, \vec{a}, 1) = a^i$. Initially, for an easier comparison with the Nash equilibrium, we assume player's are not learning the state value and transition costs either, and it is given by

$$\begin{aligned} &\phi^1(t+1, X^1_{t+1}, X^2_{t+1}) = -\mathbf{1}_{\left\{X^2_{t+1}=1\right\}}, && F^1(t, X^1_t, X^2_t, a^1, a^2) = ca^1, \ \ (c < 1) \\ &\phi^2(t+1, X^1_{t+1}, X^2_{t+1}) = -\mathbf{1}_{\left\{X^1_{t+1}\neq X^2_{t+1}\right\}}, && F^2 = 0 \end{aligned}$$

In words, the first player wants second player to move state 0, and the second player wants to be in the same state as the first player. However, the first player gains some by increasing the odds of moving to state 1. Now, costs of each player are

$$\begin{aligned} &J^1(t, \vec{x}, \vec{a}) = ca^1 - \mathbb{P}^{t,\vec{x},\vec{a}}(X^2_{t+1} = 1), \quad && J^2(t, \vec{x}, \vec{a}) = -\mathbb{P}^{t,\vec{x},\vec{a}}(X^1_{t+1} \neq X^2_{t+1}), \\ &\mathbb{P}^{t,\vec{x},\vec{a}}(X^2_{t+1} = 1) = a^2, && \mathbb{P}^{t,\vec{x},\vec{a}}(X^1_{t+1} \neq X^2_{t+1}) = a^1 + a^2(1 - 2a^1) \end{aligned} \tag{2.15}$$

From now on, as the one-step game does not depend on the current time and state, we will drop them from notations. Let us also mention that players makes decisions simultaneously. One could set it up as turn-based, but we aim to keep the game as symmetric as possible only excluding the cost structure.

Note that there is a unique Nash equilibrium when the horizon is fixed at 1, which is $\vec{a} = (1, 1)$. Although exists, it is not necessarily useful for characterizing potential behaviors of the players. Once the first player fixes the action of the second player, the player becomes unaware of the intentions of the second player. Moreover, since players do not announce their strategies, searching for a Nash equilibrium with a larger horizon does not necessarily model this game either.

Now, we will construct a learning algorithm and numerically explore how the corresponding behaviors are evolving. To do so, we need to determine what players can observe. There are two immediate choices as at the time t: (i) players can observe the action taken by the other, or (ii) players can observe only the state. Notice that (i) is simpler because state can only be 0 or 1, and does not carry all the information about the underlying action of the player. Thus, we consider (ii) from now on.

Let us recap how the game is played in the perspective of the first player. First, we determine a probability a^1 for us to transition state 1, and we get paid ca^1. Then, we loose \$1 if the other player ends up in state 1. In the perspective of the second player, objective is to simply to follow the other player. We observe the past states of the other player and try to end up in the same state, and otherwise loose \$1. Due to the simplicity of the game, we can generate strategies that are expected to be observed. Starting from the second player, as she has a simpler cost structure;

- Determine what is an acceptable level of noise in the observation of the other players state, which relies on the expectation of the player. If the recent observations are yielding a certain outcome (0 or 1) consistent with the acceptable noise level, take the corresponding action. Else, start to explore other actions with rationale to penalize the noise.

In case of the first player, cost structure is slightly more intricate;

- If the second player always appears at 0, explore larger actions to reduce cost (due to $-ca^1$). Keep increasing it until the second player starts to appear at state 1 and overcomes the gains from $-ca^1$.

- If the second player appears at 1, which is costly, switch to actions that are not yet well explored. Keep searching until the second player appears at 0 regularly enough to keep realized cost consistent with expectations.

We remark that a straightforward Q-learning algorithm can be used to model players. However, Q-learning only models expected rewards for a given action, thus it converges to fixed actions and lacks the underlying details we aim to demonstrate. This underscores the same point: it is crucial to incorporate the design of the player to understand games.

2.1.1 Details of the Learning Algorithm

We now introduce the relevant parts of the framework specific for this problem. In general, each player models a transition cost $\hat{F}$ and an estimate $\hat{\Gamma}$ on the other player's actions, relying on their observations held in the memory.

Recall that observations of players are the realized states. That is, $\mathcal{E}^1 = \mathcal{E}^2 = \{0, 1\}$, and observations in (2.10) are depending on the realizations of the states. Here, Ω^u and $\mathbb{P}^u$ are determined by the random number generators that determines the transitions of states for the players at each round. We then set the observations $\mathcal{O}^1, \mathcal{O}^2$ in equation (2.10) in an obvious manner. In the simulations, each player keeps a memory of a certain length, recording the realized states and costs.

Now, let $\{\mathcal{N}_\Gamma^{i,k}\}_{k=1}^K$ be the i'th player action networks where $\mathcal{N}_\Gamma^{i,k} : [0,1] \to [0,1]$. We then assing $\mathfrak{L}_\Gamma^i : \mathscr{E}^i \to ([0,1] \mapsto \mathcal{P}([0,1]))$ in (2.11) as the empirical distribution formed by $\{\mathcal{N}_\Gamma^{i,k}\}_k$'s. That is,

$$ {}^n\hat{\Gamma}^i := \mathfrak{L}_\Gamma^i({}^n\mathcal{O}^i) := \frac{1}{K}\sum_{k=1}^K \delta_{\mathcal{N}_\Gamma^{i,k}} $$

Note that the parameters of networks are depending on the observations, which is not explicit in notations as we view $\mathfrak{L}_\Gamma^i$ yielding the network with such parameters. To train these networks, after each step, players draw a batch of memories using the multinomial distribution with higher weights assigned to recent observations. Then, networks are getting trained to reduce the difference between estimated action and observed state.

Similarly, we denote the cost networks as $\{\mathcal{N}_F^{i,\ell}\}_{\ell=1}^K$[5], where $\mathcal{N}_F^{i,\ell} : [0,1] \to \mathbb{R}$. We then set $\hat{F} : \hat{\Omega} \times \mathbb{A} \to \mathbb{R}$ in (2.5) as

$$ \hat{F}^1(\ell, \hat{\omega}', a^1) = -ca^1 + \mathcal{N}_F^{1,\ell}(\hat{\omega}')(a^1), \quad \hat{F}^2(\ell, \hat{\omega}', a^2) = \mathcal{N}_F^{2,\ell}(\hat{\omega}')(a^2) $$

We identify $(\ell, \hat{\omega}') \in \hat{\Omega} = \{1, \dots, K\} \times \hat{\Omega}'$ where $\hat{\mathbb{P}}$ assigns the first marginal as uniform distribution over $\{1, \dots, K\}$.[6] Each network $\mathcal{N}_F^{i,\ell}$ is further random by the virtue of dropout layers. Keeping the networks always in the training mode, one generates a random function with positive dropout probabilities, and $\hat{\Omega}'$ characterizes this. Here,

$$ \mathfrak{L}_F^i : \mathscr{E}^i \to \big((\hat{\Omega}, \mathbb{A}) \mapsto \mathbb{R}\big) $$

[5] We choose the same K only for the notational simplicity.

[6] As in the case of action networks, this is only for simplicity. One might assign and evolve weights corresponding to networks, and capture more vibrant dynamics if the game is more sophisticated. For example, one might keep a subset of networks as trusted ones (high weights), and let other networks explore more wildly (low weights).

yielding ${}^n\hat{F}^i$ and ϕ^i's are taken as constant.

There are two objectives cost networks are training for; (i) there is an expected cost coming from the predictions of action networks, which is (2.15) integrated as in (2.6). If action networks are not perfect, expected cost will not match the observed expected cost. Cost networks are training to close this gap, by relying on a cost memory similar to action memory. (ii): players have expectations over what is best possible as introduced in (2.13), which we took as constant for simplicity. Cost networks are also training such that the players do not get desperate. In the simulations, if the first player ends up with networks $\mathcal{N}_\Gamma^{1,k}$'s taking values close to 1, independent of the action, they start to play the Nash equilibrium $(1, 1)$. That is, player 1 gets desperate, and then adjusts the random component of cost to start explore other actions.

We note that (2.6) becomes

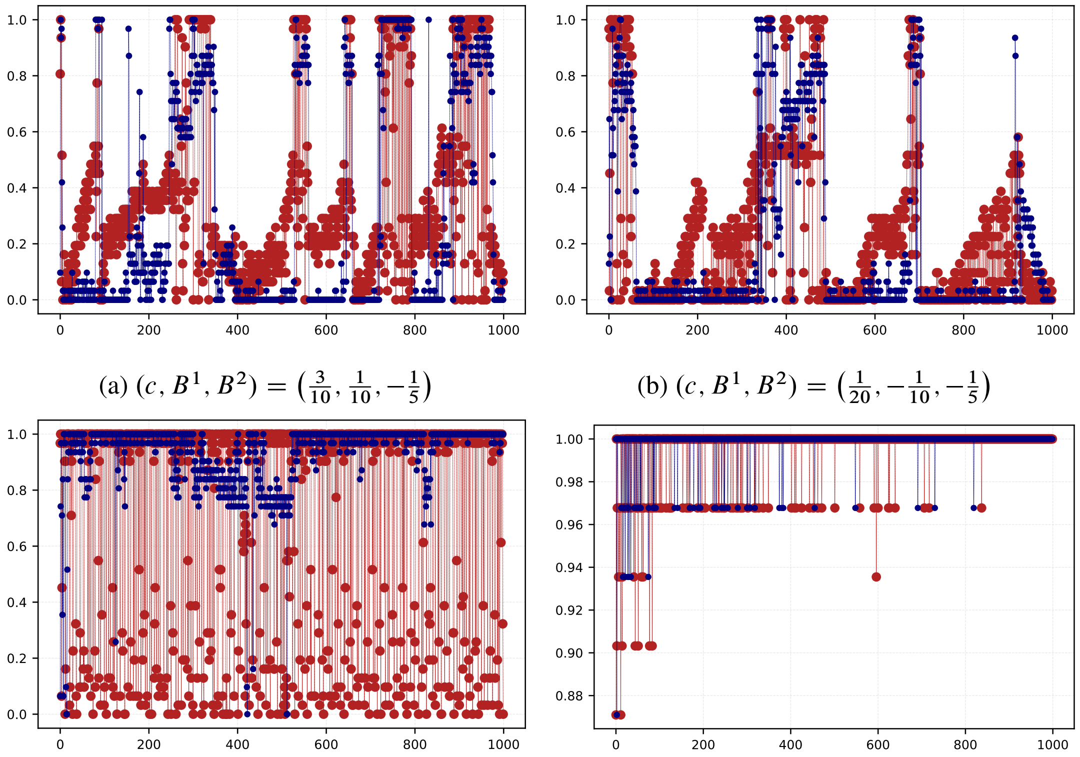
$$J^1(\ell, \hat{\omega}', a^1) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}_\Gamma^{1,k}(a^1) - ca^1 + \mathcal{N}_J^{1,\ell}(\hat{\omega}')(a^1), \quad \text{and}$$

$$J^2(\ell, \hat{\omega}', a^2) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}_\Gamma^{1,k}(a^2) + a^2(1 - 2\mathcal{N}_\Gamma^{1,k}(a^2)) + \mathcal{N}_J^{1,\ell}(\hat{\omega}')(a^2)$$

To draw an action from (2.8), players randomly choose one cost network $\ell \in \{1, \dots, K\}$, and observe one realization $\hat{\omega}'$ coming from dropout layers. Then, they simply minimize J^i over a^i, that is $\hat{\pi}$ yields deterministic ε-optimal action, and play that action.

Before discussing the simulation results, let us mention that each parameter of networks of course plays a significant role, and we coarsely tuned them by hand to obtain simulations matching with expectations. Many of such parameters are taken as constant. Changing to different constants might of course yield poor results. On the other hand, generalizing them will improve the sophistication of agents strategies. Besides the network parameters, there are more structural parameters too. For example, what players are expecting as the best possible cost also changes the characteristics of actions. Especially if the first player is expecting much better than what is realistically possible, exploration gets out of hand and they don't converge anywhere. For an another example, if the memory is very long and not forgetting, then the first player starts to get advantage over the second player, as the second player becomes fixed after a while and "thinks" that the other player will still frequently move to state 0. The point we are aiming to convey here is that the learning algorithms of players are crucial to characterize what is going to be realized, and the evolution of ${}^n\gamma^i_{\mathfrak{L}} \in \mathcal{P}([0, 1])$ might be dynamic even in the simplest settings.

Now, let us annotate the simulation results. In Figure 1, the actions taken by Player 1 (red) and Player 2 (blue) over 1000 games are plotted, with both players starting from arbitrary initial estimations. While keeping Player 2's parameters constant, four plots illustrate variations in c, the



(a) $(c, B^1, B^2) = \left(\frac{3}{10}, \frac{1}{10}, -\frac{1}{5}\right)$

(b) $(c, B^1, B^2) = \left(\frac{1}{20}, -\frac{1}{10}, -\frac{1}{5}\right)$

(c) $\left(c, B^1, B^2\right) = \left(1, 1, -\frac{1}{5}\right)$

(d) $\left(c, B^1, B^2\right) = \left(1, 0, -\frac{1}{5}\right)$

Figure 1: Actions of Player 1 (red) and Player 2 (blue) over 1000 games, demonstrating the impact of varying the incentive parameter c and expectation B^1 for Player 1, while Player 2's parameters remain constant. Each subplot shows how different incentives and expectations influence Player 1's strategy and interaction dynamics in this toy problem.

incentive parameter for Player 1, and B^1, as defined in (2.13), which represents the expectation of Player 1. Thin lines in the plots indicate the jumps between actions.

In subplot (a), Player 1 has a somewhat large incentive ($c = 3/10$) to take larger actions, aiming for a reward of $B^1 = 1/10$. Thus, playing close to $(0, 0)$ does not suffice, and Player 1 searches for higher rewards, leading to frequent changes between different phases. Notice that as soon as (a^1, a^2) approaches $(1, 1)$, Player 1 begins to explore and pushes the game back towards $(0, 0)$. In subplot (b), the incentive is much smaller ($c = 1/20$) and the expectation of Player 1 is decreased to $B^1 = -1/10$. Consequently, Player 1 stays close to $(0, 0)$ for longer, before starting to think that Player 2 will always choose action 0. In subplots (c) and (d), the incentive for Player 1 is really high, making deviations from $(1, 1)$ unnecessary. Specifically, in subplot (d), Player 1 is satisfied with a reward of 0, maintaining $(1, 1)$ almost always. Conversely, in subplot (c) where Player 1 expects to get an unrealistic reward of 1, exploration by Player 1 leads to worse results for both.

We would like to conclude this section by emphasizing again that games are inherently complex and that observed behaviors require a more detailed representation of players. We refer to the supplementary online repository [6] for the animation showing the cost networks, action networks, and other observables in each case.

3 Stated Mean Field Games

In this section, we will introduce a mean field type version of the framework. It is important to note that learning parameters are defined for each player individually. Therefore, embedding a mean field game requires adjusting the learning parameters of a representative agent to model infinitely many similar players. In particular, to align with a similar structure in the literature, we assume that essentially only $\hat{\Gamma}$ will be learned, while other parameters $(\hat{T}, \hat{p}, \hat{F}, \hat{\phi}) = (T, p, F, \phi)$ are modeled by the representative player as known (and not learned). We will thus refer to this case as the stated mean field game.

Let the state space be $\mathbb{S}_t$, $\mathbb{S} := \bigcup_{t\in\mathbb{T}} \mathbb{S}_t$, and $\mathbb{A}$ be the common action space. Set the canonical space $\Omega := \prod_{t\in\mathbb{T}} \mathbb{S}_t$ and introduce the set of controls as

$$\mathcal{A} := \left\{\alpha : \mathbb{T} \times \Omega \times \mathcal{P}(\Omega) \to \mathbb{A} \ : \ \alpha(t,x,\mu) \in \mathbb{A}, \ \forall (t,x,\mu) \in \mathbb{T} \times \mathbb{S}_t \times \mathcal{P}(\mathbb{S}_t)\right\}$$

As before, we require any function on $\mathbb{T} \times \Omega \times \mathcal{P}(\Omega)$ to be Markovian. Transition probabilities are given as

$$\begin{aligned} &p(t,x,\mu,a;y) : \mathbb{T} \times \Omega \times \mathcal{P}(\Omega) \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}^+, \quad \text{where} \\ &p(t,x,\mu,a;\cdot) \text{ is a probability measure on } \mathbb{S}_t, \text{ for all } t \in \mathbb{T}, x \in \mathbb{S}_t, \mu \in \mathcal{P}(\mathbb{S}_t) \end{aligned}$$

Next, along the idea that the representative agent is insignificant in the population, we assume $\hat{\Gamma}$ in (2.3) is constant as $\hat{\Gamma} : \mathbb{T} \to \mathcal{P}(\mathcal{P}(\mathbb{S}_t \times \mathcal{A}))$. Here, $\mathcal{P}(\mathbb{S}_t \times \mathcal{A})$ corresponds to $\vec{\mathcal{A}}$ in (2.3). In the case of countable players, indexing was keeping track of the connection between the state and the control of individiual players. Here, state variable keeps track of distribution of controls used by the population.

Now, given a particular estimation of the population $\Xi_t \in \mathcal{P}(\mathbb{S}_t \times \mathcal{A})$ by the representative player at time t, introduce $\Xi_s \in \mathcal{P}(\mathbb{S}_s \times \mathcal{A})$ recursively as

$$\Xi_{s+1}(dy, d\alpha) = \int_{\mathbb{S}_t} p(s,x,\mu^{\Xi}_s, \alpha(s,x,\mu^{\Xi}_s); dy) d\,\Xi_s(x, d\alpha), \ \forall t \leq s, \ \text{where } \mu^{\Xi}_s := \Xi_s(\cdot, \mathcal{A}) \tag{3.1}$$

Note that μ^{Ξ} corresponds to (2.4) for the population. If the second marginal of Ξ is a Dirac measure δ_α independent of the state, we call it homogeneous, as it models every individual player using a single control α. Otherwise, we call it heterogeneous. In the homogeneous case, we do not need to keep track of the flow of the distribution of controls. Moreover, in the heterogeneous case, one can represent the flow of the population μ^{Ξ} using a single relaxed control instead of a distribution of controls. See [5] for the details.

Introduce the flow of the distribution for the representative player;

$$\mathbb{P}^{t,\Xi;x,\alpha}(X_{s+1} = dy | X_s = \tilde{x}) = p(s, \tilde{x}, \mu_s^{\Xi}, \alpha(s, \tilde{x}, \mu_s^{\Xi}); dy) \ \ \forall t \leq s, \ \ \mathbb{P}^{t,\Xi;x,\alpha}(X_t = x) = 1 \tag{3.2}$$

where X is the canonical process. Notice that the player is observing the distribution of the population μ^{Ξ}, given the initial data $\Xi \in \mathcal{P}(\mathbb{S}_t \times \mathcal{A})$.

Recall that we assume the cost is known and not learned. Moreover, while defining (2.6), we started from the initial state x, and here we similarly start from the current distribution $\mu \in \mathcal{P}(\mathbb{S}_t)$. We will restrict the learning algorithm to yield $\hat{\Gamma}$ with its marginal on $\mathbb{S}_t$ as a Dirac measure at μ. Then, similar to (2.6), we assume the cost structure is given by

$$\begin{aligned} J(t, \mu; x, \alpha) &:= \int_{\mathcal{P}(\mathbb{S}_t \times \mathcal{A})} J(t, \Xi; x, \alpha) d\hat{\Gamma}_t(\Xi), \ \text{ where } \ \hat{\Gamma}_t((\mu, \mathcal{P}(\mathcal{A}))) = 1, \ \text{and} \\ J(t, \Xi; x, \alpha) &:= \mathbb{E}^{t,\Xi;x,\alpha}\Big[\phi(X_{t+T}, \mu_{t+T}^{\Xi}) + \sum_{s=t}^{t+T-1} F(s, X_s, \mu_s^{\Xi}, \alpha(s, X_s, \mu_s^{\Xi}))\Big], \quad \mathbb{E}^{\cdot} := \mathbb{E}^{\mathbb{P}^{\cdot}} \end{aligned} \tag{3.3}$$

Set $\mathcal{M}_J$ as the space of all such functions $(\mathbb{T} \times \mathcal{P}(\Omega) \times \Omega \times \mathcal{A} \to \mathbb{R})$. Let us note that, we are mainly interested in the static $\{0, \ldots, T\}$ problem for simplicity. One can dynamically set $\hat{T} = T - t$ (and repeats after T) by the learning algorithm to create a dynamic version. Or one might evolve the game indefinitely, keeping the $\hat{T}$ fixed.

Given the cost, we now need to estimate the optimal controls by the learning parameter

$$\hat{\pi} : \mathbb{T} \times \mathcal{P}(\Omega) \times \Omega \to \mathcal{P}(\mathcal{A})$$

There is no randomness in the value, and hence if the representative player is able to solve for the optimal control, $\hat{\pi}$ becomes deterministic, or in general, takes values on the set of optimal controls. Moreover, induced distribution is simply $\Upsilon^{t,\mu;x}(d\alpha) := \hat{\pi}(t, \mu, x)$ and

$$\gamma^{t,\mu;x}(da) := \Upsilon^{t,\mu;x}\Big(\alpha \ : \ \alpha(t, x, \mu) = da\Big) \in \mathcal{P}(\mathbb{A})$$

Finally, we are ready to introduce the crucial part: observations and the learning algorithm. Let us briefly recap the mean field framework. We assume that the representative player starts with an initial guess of the population distribution over states and controls, determines the corresponding optimal control, and, relying on the assumption that everyone else is exactly the same, generates further observations using the chosen learning algorithm. To formalize this, let $\mathcal{E} = \mathcal{P}(\mathcal{P}(\mathbb{S}\times\mathcal{A}))$ be the space of observables, and let $\mathscr{E}$ denote the space of finite sequences of $\mathcal{E}$. Recall that the learning algorithm is a mapping $\mathfrak{L}_\varphi : \mathscr{E} \to \mathcal{M}_\varphi$, where the player's estimation at age n is ${}^n\mathfrak{L}_\varphi := \mathfrak{L}_\varphi({}^n\mathcal{O})$, and the player's prior is ${}^0\mathfrak{L}_\varphi$. Here, $\mathcal{O} : \Omega^u \times \mathbb{N} \to \mathscr{E}$ represents the increasing sequence of observations. The set Ω^u corresponds to a potential random choice made by the player during the search for optimal control. To provide intuition, we will construct a simple but explicit learning algorithm for $\mathfrak{L}_\Gamma$:

Suppose that the current distribution of the population at time 0 is $\mu \in \mathcal{P}(\mathbb{S}_0)$ and is fixed as given. We, as the representative player, start with an initial guess ${}^0\mathcal{O} = {}^0\Xi = \delta_{(\mu,\delta_{{}^0\alpha})}$ for some ${}^0\alpha$. That is, our initial observation is $\delta_{{}^0\Xi}$. Then, we determine the population flow $\mu^{{}^0\Xi}$ using (3.1), our flow $\mathbb{P}^{t,{}^0\Xi;x,\tilde{\alpha}}$ using (3.2), and solve the optimization problem to find an optimal control ${}^1\alpha \in \operatorname{supp}\Upsilon^{0,\mu;x}$.

Now, following the fixed point idea, we learned that if the population is using ${}^0\alpha$, it is optimal to use ${}^1\alpha$. Since every player is equivalent, we may deduce that the population will use ${}^1\alpha$ with some probability c, and use ${}^0\alpha$ otherwise. That is, the learning algorithm yields

$$\mathfrak{L}_\Gamma(\delta_{(\mu,\delta_{{}^0\alpha})}) = c\delta_{(\mu,\delta_{{}^1\alpha})} + (1-c)\delta_{(\mu,\delta_{{}^0\alpha})} = {}^1\mathcal{O}$$

Notice that, for simplicity, we are assuming a homogeneous population. That is, everyone is assumed to be using a single control. Now, we can repeat the same procedure to find another optimal control under the guess $\hat{\Gamma} = \mathfrak{L}_\Gamma(\delta_{(\mu,\delta_{{}^0\alpha})})$, denoted as ${}^2\alpha$, and so on. In general, our naive learning algorithm depends only on the last observation, defined as

$$\mathfrak{L}_\Gamma\left({}^n\mathcal{O}\right) := c\ \delta_{(\mu,\delta_{{}^{n+1}\alpha})} + (1-c)\ {}^n\mathcal{O},\quad 0 \le c \le 1 \tag{3.4}$$

where we took ${}^n\mathcal{O} \in \mathcal{P}(\mathcal{P}(\mathbb{S}_0 \times \mathcal{A}))$ as a single observation rather than a sequence to simplify notation, and ${}^{n+1}\alpha$ is an optimal control under ${}^n\mathcal{O}$. We are assuming that $\hat{\pi}$ yields exact optimal controls, and we do not explicitly account for the possibility of multiple optimal controls. There might be a deterministic selection, or it could be randomly selected. In the latter case, $\mathbb{P}^u$ characterizes this random choice.

Let us remark on the similarity between the fictitious play-type algorithms introduced in [4]. In fictitious play, one also starts with an initial guess $\delta_{{}^0\alpha}$ and finds the optimal ${}^1\alpha$. A crucial difference, however, is that fictitious play considers the weighted average of $\mu^{{}^0\Xi}$ and $\mu^{{}^1\Xi}$ to find the next optimal control α^2. That is, the cost structure becomes

$$J^{\text{fictitious}}(t,\mu;x,\alpha) := J\left(t, \int_{\mathcal{P}(\mathbb{S}_t\times\mathcal{A})} \Xi\, d\hat{\Gamma}(\Xi); x, \alpha\right)$$

for J as in (3.3), with the $\hat{\Gamma}$ induced by the same $\mathfrak{L}_\Gamma$ but solving a different optimization. We leave the question of whether these approaches are equivalent for potential games to future research, which is a key assumption in [4] for the convergence result.

Lastly, we rephrase the definition of uncertain equilibrium and will explicitly compute the equilibrium under this basic learning algorithm in the next section.

Definition 4 (Uncertain Equilibria of stated Mean Field Games) *We say $(\hat{\Gamma}, \hat{\pi}) \in (\mathcal{M}_\Gamma, \mathcal{M}_\pi)$ is an (ε, r, δ)-uncertain equilibrium at $(t, x, \mu) \in \mathbb{T} \times \mathbb{S}_t \times \mathcal{P}(\mathbb{S}_t)$ under the learning algorithm $(\mathfrak{L}_\Gamma, \mathfrak{L}_\pi)$ if,*

(i) $(\hat{\Gamma}, \hat{\pi})$ *is the prior of the player,*

(ii)

$$\sup_{\alpha \in \mathcal{A}} {}^nJ(t, \mu; x, \alpha) - \int_{\mathcal{A}} {}^nJ(t, \mu; x, \alpha)^n\hat{\pi}(t, \mu, x)(d\alpha) \le \varepsilon, \quad \forall n \in \mathbb{N}$$

(iii)

$$\mathbb{P}^u\Big(\liminf_{n\to\infty} d^{t,\mu;x}\big({}^0\Upsilon^{t,\mu;x}_{\mathfrak{L}}, {}^n\Upsilon^{t,\mu;x}_{\mathfrak{L}}\big) > r\Big) \le \delta$$

where $d^{t,\mu;x}$ is the metric the player equips the space $\mathcal{P}(\mathcal{A})$ under the equivalence $=^{t,\mu;x}$.

3.1 One step stated mean-field game examples

We now present two examples in which we can explicitly compute and contrast the relaxed equilibrium and uncertain equilibrium under the learning algorithm described in (3.4). In the first example, while there is no standard Nash equilibrium, a relaxed equilibrium does exist. Conversely, in the second example, due to the cost function being discontinuous, there is no relaxed equilibrium; however, the uncertain equilibrium remains unchanged.

Example 1 *Set $\mathbb{S} = \{0, 1\}$, $\mathbb{T} = \{0, 1\}$, the action space $\mathbb{A} = [0, 1]$, and the transition probability*

$$p(0, x, a, \mu; 1) = a, \qquad p(0, x, a, \mu; 0) = 1 - a$$

Furthermore, introduce the cost as

$$J(\Xi; \alpha) := \mathbb{E}^{\mathbb{P}^{\Xi,\alpha}}\Big[\phi(X_1, \mu_1^{\Xi}) + F(\alpha)\Big]$$
$$\text{where} \quad \phi(x, \mu) := 2|\mu(1)|^2 - 4\mathbf{1}_{\{x=1\}}\mu(1), \ \text{ and } \ F(a) = (|a|^2 + a)$$

Then, there exists no standard Nash equilibrium and a unique relaxed equilibrium $\frac{1}{2}(\delta_0 + \delta_1)$.The learning algorithm described in (3.4) oscillates around $\frac{1}{2}(\delta_{\delta_0} + \delta_{\delta_1}) \in \mathcal{P}(\mathcal{P}(\mathbb{A}))$, and induces an action distribution δ_0 or δ_1 infinitely often.

Proof First, let us argue that there exists no standard Nash equilibrium. Main idea is, if the population distribution is symmetric $\mu_1^\Xi(1) = \mu_1^\Xi(0) = 1/2$, then the optimal actions are $\{0, 1\}$. Whenever $\mu_1^\Xi(1) > 1/2$, optimal action becomes 0 and otherwise 1. That is, every player tries to stay away from the majority and there cannot be a deterministic fixed point.

As for the standard Nash equilibrium population is homogeneous, (3.1) simplifies to

$$\mu^a(1) := \mu_1^\Xi(1) = a$$

whenever the population is taking the action $a \in \mathbb{A}$, independent of the initial distribution. For the representative player, we reserve $\tilde{a} = \alpha(0, x)$ and compute the cost;

$$\begin{aligned} J(a, \tilde{a}) := J(\Xi; \alpha) &= 2|\mu^a(1)|^2 - 4\mu^a(1)\mathbb{P}^{\Xi,\alpha}(X_1 = 1) + |\tilde{a}|^2 + \tilde{a} \\ &= 2a^2 - 4a\tilde{a} + |\tilde{a}|^2 + \tilde{a} \end{aligned}$$

since it is quadratic in $\tilde{a}$, maximum occurs only if $\tilde{a} \in \{0, 1\}$. Thus, noting that

$$J(0, \tilde{a}) = |\tilde{a}|^2 + \tilde{a}, \quad J(1, \tilde{a}) = 2 + |\tilde{a}|^2 - 3\tilde{a}$$

there exists no standard Nash equilibrium.

Now, to compute the relaxed equilibrium, we know from [5] that it is equivalent to consider the heterogeneous case. Thus, as the initial distribution is irrelevant, let $\Xi_0 \in \mathcal{P}(\mathcal{A}) = \mathcal{P}(\mathbb{A})$. Since there is only a single time step, the distribution of controls doesn't evolve either and (3.1) becomes

$$\Xi(1, da) := \Xi_1(1, d\alpha) = a\Xi_0(da), \qquad \mu^\Xi(1) := \mu_1^\Xi(1) = \Xi(1, \mathbb{A}) = \int_{[0,1]} a\Xi_0(da)$$

Then,

$$J(\Xi_0; \tilde{a}) := J(\Xi; \alpha) = 2|\mu^\Xi(1)|^2 - 4|\mu^\Xi(1)|\tilde{a} + |\tilde{a}|^2 + \tilde{a}$$

and in this case, again from [5], equilibrium means that every action in the support of Ξ_0 is optimal. It is easy to check that if $\mu^\Xi(1) \neq 1/2$, then the optimal action is either 0 or 1 and there cannot be any equilibrium. If $\mu^\Xi(1) = 1/2$, then both 0 and 1 are optimal. Thus, $\Xi_0 = \frac{1}{2}(\delta_0 + \delta_1)$ corresponds to the relaxed equilibrium, since any action in the support is optimal.

Lastly, let us discuss the convergence of (3.4). Consider any $\Gamma = \sum_i c_i \delta_{\delta_{ai}}$ which is an element of $\mathcal{P}(\mathcal{P}(\mathbb{A}))$ if $\sum_i c_i = 1$ representing any homogeneous estimate for the action of the population. Then, under appropriate notational simplifications of this example, (3.3) becomes

$$J(\tilde{a}) = \int_{\mathcal{P}(\mathbb{A})} J(\Xi, \tilde{a}) d\Gamma(\Xi) = \sum_i c_i J(\delta_{a_i}, \tilde{a}) = 2\sum_i c_i |a_i|^2 - 4\tilde{a}\sum_i c_i a_i + |\tilde{a}|^2 + \tilde{a}$$

which is exactly as before a quadratic polynomial in $\tilde{a}$, hence optimal value occurs at either 0 or 1. Therefore, the algorithm (3.4) will quickly converge to a distribution having $\delta_0, \delta_1 \in \mathcal{P}(\mathbb{A})$

in its support, and the contribution from the initial guess will diminish with the factor $(1-c)^n$. Moreover, as the optimal $\tilde{a}$ becomes 0 or 1 depending on the estimated average of the population similarly as before, the algorithm will oscillate around $\frac{1}{2}(\delta_{\delta_0}+\delta_{\delta_1})$. Note that, by adjusting the constant parameter c in (3.4), one can achieve exact convergence too[7]. Here, since $\hat{\pi}$ is computed exactly as either δ_0 or δ_1, we also observe that the induced distribution oscillates in between them infinitely often.

$\blacksquare$

Example 2 *Set* $\mathbb{S}=[0,1]$, $\mathbb{T}=\{0,1\}$, *the action space* $\mathbb{A}=[0,1]$, *and the transition probability*

$$p(0,x,a,\mu;dy)=\delta_a$$

Furthermore, introduce a discontinuous cost as

$$J(\Xi;\alpha):=\mathbb{E}^{\mathbb{P}^{\Xi;\alpha}}\left[X_1^\alpha\mathbf{1}_{\left\{\bar{\mu}_1^\Xi\in\left[0,\frac{1}{2}\right]\right\}}-X_1^\alpha\mathbf{1}_{\left\{\bar{\mu}_1^\Xi\in\left(\frac{1}{2},1\right]\right\}}\right]$$

where $\bar{\mu}^\Xi := \int_{[0,1]} x\,d\mu^\Xi$. *Whereas no relaxed equilibrium exists, the learning algorithm described in* (3.4) *again oscillates around* $\frac{1}{2}(\delta_{\delta_0}+\delta_{\delta_1})\in\mathcal{P}(\mathcal{P}(\mathbb{A}))$, *and induces an action distribution* δ_0 *or* δ_1 *infinitely often.*

Proof The essence of this example is similar to that in Example 1. It is immediately clear that there exists no relaxed equilibrium, as the optimal action is either δ_0 or δ_1 under any value of $\bar{\mu}_1^\Xi$, and neither constitutes an equilibrium.

For the learning algorithm (3.4), although the cost function is computed differently than in Example 1, $\hat{\pi}$ behaves exactly the same, depending on the population average. Thus, there is no difference from Example 1.

$\blacksquare$

4 Reinforcement Learning

In this section, we consider the special case of the framework presented in Section 2 when applied to a single player, thereby reducing the problem to a control setting. The main objective is to demonstrate that our framework generalizes Markov Decision Processes, and consequently, well-established methods from the reinforcement learning (RL) literature can be naturally incorporated.

[7] Let us briefly take attention to the importance of the design of the learning algorithm, even for this simple setting. If the parameter c diminishes very fast with n, then one can conclude either δ_0 or δ_1 is optimal depending on the initial condition.

Let us first note that the standard RL terminologies correspond to the learning parameters we have introduced. Learning $\hat{p}$ is typically carried out using model-based methods. In its simplest form, learning the horizon $\hat{T}$ is achieved by adjusting the discount factor[8]; in more advanced frameworks, it is modeled as a stopping time and incorporated within the options framework. The parameter $\hat{F}$, which represents the rewards, may either be predefined or learned through various methods, such as inverse reinforcement learning. The function $\hat{\phi}$ plays a similar role to the standard value function and is considered distributional RL because it is treated as a random variable. Learning $\hat{\pi}$ is typically referred to as policy learning, while $\hat{\Gamma}$ is sometimes called behavior prediction. It is important to note that an agent may employ more sophisticated strategies by modeling additional, related learning parameters. For example, this could include learning to encode raw observations, establishing communication protocols, understanding opponents' incentives, or tuning exploration parameters, such as the best expected rewards in the two-player game example in Section 2.1.

To simplify the discussion and notation, we assume that the state and action spaces, $\mathbb{S}$ and $\mathbb{A}$, are fixed and that the estimated functions are time-invariant. In the absence of other players, $\hat{\Gamma}$ becomes irrelevant. The observations ${}^{n}\mathcal{O}$, which might include states, actions, and rewards, are generated by the environment (or universe) $(\Omega^u, \mathcal{F}^u, \mathbb{P}^u)$ in which the player operates. The player's objective is to learn $(\hat{T}, \hat{p}, \hat{F}, \hat{\phi}, \hat{\pi})$. Given $(\hat{T}, \hat{p}, \hat{F}, \hat{\phi})$, we have the cost function as

$$J(x;\alpha) := \mathbb{E}^{\mathbb{P}^{x,\alpha}}\Big[\hat{\phi}(X_{\hat{T}}) + \sum_{t\geq 0}^{\hat{T}-1} \hat{F}(X_t, \alpha(t, X_t))\Big]$$

Then, the player aims to find the optimal controls of $J(x;\alpha)$ using $\hat{\pi} \in \mathcal{P}(\mathcal{A})$, which yields induced distributions Υ^x, γ^x as in (2.8), (2.9). We remark that, instead of treating $\hat{F}, \hat{\phi}$ as known and then assigning a method of randomizing actions to explore, we model $\hat{F}, \hat{\phi}$ as unknown, which then already induces a distribution over actions. This aligns well with the intuition that better characterized values should yield less uncertain strategies.

Let us observe how we can mimic the Markov Decision Processes. Let $(X, \mathcal{I})$ be the canonical process on $(\mathbb{S} \times \mathbb{A})^\infty$. Given $(x_0, a_0) \in \mathbb{S} \times \mathbb{A}$, introduce an induced distribution from the model of the player $\hat{p}$ and γ^x as

$$\begin{cases} \tilde{\mathbb{P}}^{x_0,a_0}(X_{s+1} = dy, \mathcal{I}_{s+1} = d\tilde{a} | X_s = x, \mathcal{I}_s = a) := \hat{p}(x, a; dy)\gamma^y(d\tilde{a}), \quad s \geq 0 \\ \tilde{\mathbb{P}}^{x_0,a_0}(X_0 = x_0, \mathcal{I}_0 = a_0) = 1 \end{cases}$$

Now, we can introduce a related Q-function,

$$Q(x,a) := \mathbb{E}^{\tilde{\mathbb{P}}^{x,a}}\Big[\sum_{t\geq 0} \lambda^t \hat{F}(X_t, \mathcal{I}_t)\Big]$$

[8] We note that considering an infinite time horizon with exponentially decaying bounded rewards is essentially equivalent to a finite time problem without considering state values.

For notation, we didn't keep track of learning index n, but both $\tilde{\mathbb{P}}$ and $\hat{F}$ (and consequently Q) are evolving as we learn. Note that Q satisfies a Bellman type relation

$$Q(x,a) = \hat{F}(x,a) + \lambda \int_{\mathbb{S}\times\mathbb{A}} Q(y,\tilde{a})\ \hat{p}(x,a;dy) \times \gamma^{y}(d\tilde{a})$$

Now, if we knew exactly what $p = \hat{p}$, $F = \hat{F}$ are, and were able to compute the optimal controls for the infinite horizon, then γ^{x} would have support on the maximizers, and $V(x) := \sup_{a} Q(x,a) = \int_{\mathbb{A}} Q(x,\tilde{a})\gamma^{x}(d\tilde{a})$. That is, we recovered the standard Bellman equation:

$$V(x) = \sup_{a}\left\{F(x,a) + \lambda \int_{\mathbb{S}} V(y)\ p(x,a;dy)\right\}$$

As usual, however, it is almost never feasible to know these parameters in reality, nor do they truly exist and fully represent the system at hand. We remark that,

$$\int_{\hat{\Omega}}\left(\int_{\mathcal{A}} J(\hat{\omega},x;\alpha)\hat{\pi}(\hat{\omega},x)(d\alpha|\alpha(t,x)=a)\right)d\hat{\mathbb{P}}(\hat{\omega})$$

assigns a scalar value for (x,a), which is better suited in our framework as it does not introduce simplifications. We also note that, in simpler settings, it might be more appropriate to learn an estimate $Q(x,a)$ directly, and avoid all the estimations we have introduced. This is of course a design choice for the player.

We will briefly illustrate a simple method for learning CartPole [7], aiming to further highlight the connections with well-known methods, and learn $\hat{\phi}$ with a fixed horizon $T = 8$. Let us revisit the basics of the CartPole problem. The state space is $\mathbb{S} = \mathbb{R}^4$ representing position, velocity, angle, and angular velocity. The action space is $\mathbb{A} = \{0,1\}$, where the actions represent applying force to the left or right of the cart. The player gains a reward at each time step if the pole is vertical within some bounds.

There is a memory of player for recording observations: states, actions and a metric evaluating the performance of the player after each episode. We took this performance metric as an average of both relative and absolute performance.

In this problem, $\hat{p}$ is deterministic and can be learned. However, as it is not of our interest, we took it as given. We also set $\hat{F} = 0$ to only model the state values $\hat{\phi}$. Introduce $\{\mathcal{N}_{\phi}^{k}\}_{k=1}^{K_\phi}$ for some $K_\phi \in \mathbb{N}$ as neural networks $\mathbb{S} \to \mathbb{R}_+$ for state values. Let $\mathfrak{L}_\phi : \mathcal{E} \to (\hat{\Omega} \times \mathbb{S} \to \mathbb{R}_+)$

$$\mathfrak{L}_\phi({}^{n}\mathcal{O}) := \sum_{k=1}^{K_\phi} \delta_{\mathcal{N}_{\phi}^{k}} \mathbf{1}_{\{k\}}(\hat{\omega}), \quad \hat{\omega} \in \hat{\Omega} := \{1,\ldots,K_\phi\}, \text{ and } \hat{\mathbb{P}} \text{ uniform}$$

Here, $\hat{\mathbb{P}}$ taken as uniform means that the player has no information about which network is providing better estimations, which is again for simplicity only. We took the range of each $\mathcal{N}_{\phi}^{k}$ as $[0,100]$, and trained these networks relying on the memory. The basic idea is that networks are trained to map the states to their corresponding performance values. Here, we randomly sample from best and worst performances in a balanced way, creating a diversity between $\mathcal{N}_\phi^k$'s. Moreover, we also trained networks to promote time-consistency.

To make sure that we are training value networks properly [9], we consider the set of all controls $\mathbb{A}^T = \{0,1\}^T$ and index them by $\{\alpha^k\}_{k=1}^{2^T}$. Then, we let $\mathfrak{L}_\pi : \mathcal{E} \to (\hat{\Omega} \times \mathbb{S} \to \mathcal{P}(\mathbb{A}^T))$ as

$$\begin{aligned} \mathfrak{L}_\pi &:= \frac{1}{Z} \sum_{k=1}^{2^T} \exp\big(J(\hat{\omega}, x, \alpha^k)\big) \delta_{\alpha^k}, \qquad (\hat{\omega}, x) \in (\hat{\Omega}, \mathbb{S}) \\ \text{where } \ & J(x;\alpha) := \mathbb{E}^{\mathbb{P}^{x,\alpha}}\big[\hat{\phi}(X_T)\big] \end{aligned}$$

and Z is the normalizing constant. Lastly, given $(\hat{\phi}, \hat{\pi})$ as $(\mathfrak{L}_\phi, \mathfrak{L}_\pi)({}^n\mathcal{O})$, the induced distribution in (2.8) is simply

$$\Upsilon^x := \int_{\hat{\Omega}} \frac{1}{Z} \sum_{k=1}^{2^T} \exp\big(J(\hat{\omega}, x, \alpha^k)\big) \delta_{\alpha^k} \, d\hat{\mathbb{P}}(\hat{\omega}) = \frac{1}{K_\phi Z} \sum_{\ell,k=1}^{K_\phi, 2^T} \exp\big(J(\ell, x, \alpha^k)\big) \delta_{\alpha^k}$$

Lastly, we will briefly examine the simplest case of Bandit Games and demonstrate how it can be embedded into our framework in two different ways. The first approach adopts a single-player perspective, where the arms are represented by the state X. The second approach treats arms as players who take actions but do not engage in learning. Both versions will help to explore our framework further, and they may generalize in different directions.

In the state perspective, let $\mathbb{S} = (S^1, \ldots, S^k)$ represent a fixed k dimensional state space. Here, transition probabilities are defined by $(\mu^1, \ldots, \mu^k)$, which are independent of the current state, time and actions, representing the simplest dynamics of each arm. Let us note that random samplings from these distributions characterize $\mathbb{P}^u$, the universal probability distribution. In this perspective, the state of the chosen arm corresponds to the reward of the main agent, and the learning objective is $\hat{p}$ aiming to learn $(\mu^1, \ldots, \mu^k)$.

Fix $\hat{T} = 1$, and $\mathbb{A} = \{1, \ldots, k\}$. As the reward of the agent depends on the current action and the next state, introduce

$$\begin{aligned} & J(\hat{\omega}, a) := \mathbb{E}^{\mathbb{P}}\big[\phi(a, X_1)\big] + \hat{F}(\hat{\omega}, a), \\ & \text{where } \ \phi : \mathbb{A} \times \mathbb{S} \to \mathbb{R} \ \text{ defined as } \ \phi(a, x) = x_{|a}, \ a\text{'th component of the state} \end{aligned}$$

Note that, we adapted the framework in a way ϕ does not necessarily corresponds to the standard value function, and $\hat{F}$ aims to capture the underlying distribution of the state.

[9] Note that, it is considerably easier to learn controls in this simple setting since $\mathbb{A}^T$ is finite, and we observed the player performs quite well even without properly trained value networks.

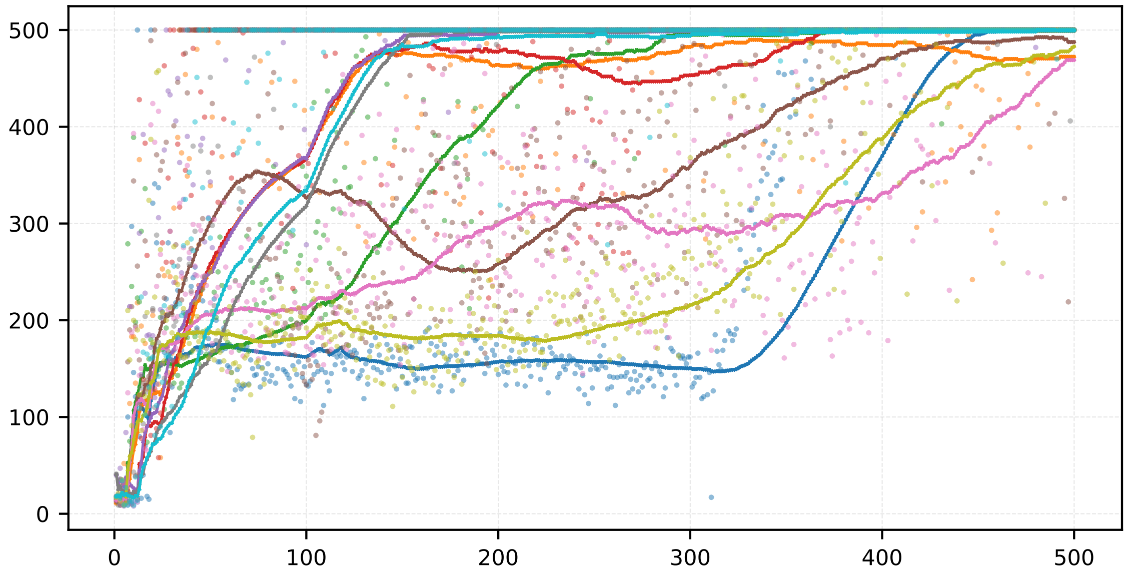


Figure 2: Performance of the CartPole Game Across 10 Selected Runs. The x-axis represents the number of games (or episodes), while the y-axis shows the total reward for each episode. Each line corresponds to one of the best 10 runs out of 32, with small markers indicating individual episode scores and solid lines showing the moving average calculated from up to the last 100 episodes.

Observe that, if the expected reward of one arm is high enough such that the randomness of $\hat{F}$ has support smaller than the difference between their expected rewards, then the optimal induced action becomes a Dirac distribution. However, typically (2.9) induces a non-degenerate distribution over actions, as it is not known by the agent which arm is definitively better. Note that determining the optimal action per event $\hat{\omega}$ is trivial, hence we do not discuss $\hat{\pi}$. Additionally, $\hat{F}$ might aim to capture some related rewards rather than the underlying distribution. For example, in the case of two arms with identical expected rewards, the agent might aim to capture different qualitative differences between the distributions.

It is important to note that an adversary can be included as another player within the model. The adversary might observe the actions of the agent, and the adversary's actions can alter the underlying transition probabilities of the states. This introduces potential extensions, such as scenarios where both the agent and the adversary are attempting to estimate each other's actions.

In the action perspective, let players $1, 2, \ldots, k$ represent arms in the game. Each arm has a fixed horizon $\hat{T} = 1$, and transition probabilities or estimating others' actions are not relevant for the arms. We characterize the reward of the main agent as the action taken by the chosen arm. For the rewards of the arms, one can define $\hat{F}^i(\hat{\omega}, a)$ on a probability space $(\hat{\Omega}, \hat{\mathbb{P}})$ such that the induced distributions over actions correspond to the targeted reward distribution μ^i. Note that, as the arms do not engage in learning (constant $\mathfrak{L}_\varphi$), they always satisfy the conditions of uncertain equilibrium.

Here, the objective of the main agent is to characterize $\hat{\Gamma}$ as defined in (2.3). Let $\hat{\Gamma}$ be independent of the agent's actions, $\hat{\Gamma} \in \mathcal{P}(\mathbb{A}^k)$, where $\mathbb{A}$ represents the action space of the arms (which corresponds to the rewards of the agent). Consequently, we can similarly define the cost function for the main agent as

$$J(\hat{\omega}, \ell, a_1, \ldots, a_k) := a_\ell + \hat{F}(\hat{\omega}, \ell), \quad \text{and} \quad J(\ell) := \int_{\mathbb{A}^k} J(\ell, a_1, \ldots, a_k) d\hat{\Gamma}(a_1, \ldots, a_k)$$

The cost function is exactly the same as in the state perspective; however, it is represented as an expectation over $\hat{\Gamma}$ rather than $\hat{p}$.

In this perspective, it is more natural to consider each arm as an individual adversary. These adversaries can cooperate, as cooperation might be manifested through $\hat{\Gamma}^i$s that estimate the actions of other adversaries. Furthermore, the main agent might aim to capture the correlated behaviors of the arms.

We stress that we have not designed any learning algorithm that constitutes an uncertain equilibrium, nor have we characterized its stability, regret bounds, or other related metrics. In the simplest scenario, where there is one optimal arm and the reward distributions are fixed, any reasonable algorithm that successfully identifies this arm would correspond to an uncertain equilibrium. However, in more complex and realistic scenarios, the specific setup must be further explored to develop a favorable learning algorithm, which falls outside the scope of this work.

References

[1] Bonesini O., Campi L., Fischer M., *Correlated Equilibria for Mean Field Games wth Progressive Strategies*

[2] Campi L., Fischer M., *Correlated Equilibria and Mean Field Games: A Simple Model*, Mathematics of Operations Research.

[3] Campi L., Cannerozzi F., Fischer M., *Coarse correlated equilibria for continuous time mean field games in open loop strategies.*

[4] Cardaliaguet P., Hadikhanloo S., *Learning in mean field games: The fictitious play*, ESAIM: COCV, 23 (2017), no. 2, 569–591. DOI: https://doi.org/10.1051/cocv/2016004.

[5] İşeri M. and Zhang J., *Set values for mean field games*, Trans. Amer. Math. Soc. 377 (2024).

[6] İşeri M., *Two Player Game*, GitHub repository, https://github.com/melihiseri/TwoPlayerGame.

[7] İşeri M., *CartPole Toy Model*, GitHub repository, available at https://github.com/melihiseri/CartPole_ToyModel.

[8] Karnam C., Ma J., Zhang J., *Dynamic approaches for some time-inconsistent optimization problems*, The Annals of Applied Probability, Vol. 27, No. 6, 2017.

[9] Muller P. et al., *Learning Correlated Equilibria in Mean-Field Games*

[10] Roughgarden T., *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press.