Pessimism Meets Invariance: Provably Efficient Offline Mean-Field Multi-Agent RL

Anonymous Author(s)
Affiliation
Address
email

Abstract

Mean-Field Multi-Agent Reinforcement Learning (MF-MARL) is attractive in the applications involving a large population of homogeneous agents, as it exploits the permutation invariance of agents and avoids the curse of many agents. Most existing results only focus on online settings, in which agents can interact with the environment during training. In some applications such as social welfare optimization, however, the interaction during training can be prohibitive or even unethical in the societal systems. To bridge such a gap, we propose a SAFARI (peSsimistic meAn-Field vAlue iteRatIon) algorithm for off-line MF-MARL, which only requires a handful of pre-collected experience data. Theoretically, under a weak coverage assumption that the experience dataset contains enough information about the optimal policy, we prove that for an episodic mean-field MDP with a horizon H and N training trajectories, SAFARI attains a sub-optimality gap of $\mathcal{O}(H^2 d_{\text{eff}}/\sqrt{N})$, where d_{eff} is the effective dimension of the function class for parameterizing the value function, but independent on the number of agents. Moreover, we also propose a variant of our SAFARI algorithm for the online setting, which is of independent interest.

1 Introduction

Significant progress has been made towards multi-agent reinforcement learning (MARL) for many prominent sequential decision making problems, such as social welfare optimization [1], fleet control of autonomous vehicles [2] and playing multiplayer online battle arena (MOBA) games [3]. As the joint state and action space scales exponentially with the number of agents, however, MARL becomes computationally expensive. One remedy is the mean-field regime when an extremely large number of homogenous agents are involved, e.g., social welfare optimization. The effect of each agent on the overall multi-agent system can become infinitesimal, and therefore all agents can be considered interchangeable/indistinguishable [4–6]. Accordingly, the interaction among agents can be captured by some mean-field quantity such as the empirical distribution of states, and therefore each agent only needs to find the best response to the so-called "mean-field state", which avoids the curse of many agents.

Most existing results on mean-field MARL (MF-MARL) are for the online setting [4, 7], where the agents can interact with the environment during training. However, such interaction during training can be prohibitive for some important applications [1, 8–10]. Taking social welfare optimization as an example, repeatedly conducting social experiments on human being can be unaffordable or even unethical in the societal systems. Therefore, we can only consider the offline settings, i.e., we learn the optimal policy based on some pre-collected experience data [10]. Unfortunately, existing offline reinforcement learning (RL) algorithms and theories all focus on the single agent settings, and no algorithms and theories have been developed for MARL under the offline settings, regardless of the mean-field regime or not.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

To bridge such a critical gap, we propose the first pessimistic algorithm – named SAFARI (peSsimistic meAn-Field vAlue iteRatIon) for mean-field MARL, which can provably achieve sample efficiency under the offline setting. Our proposed algorithm contains two important components: (1) To incorporate the permutation invariance of the homogenous agents, we adopt a RKHS (Reproducing Kernel Hilbert Space) mean-embedding approach for approximating value functions, which avoids the exponential blowup of the agents' state and action spaces; (2) We develop an uncertainty quantifier, and integrate it into the value iteration procedure as the penalty function. Such a penalty function can effectively screen the "spuriously correlated trajectories", i.e., which possibly happen to appear in the experience data, but are actually unrelated to the optimal policy, but by chance induce large cumulative rewards and hence may potentially mislead the learned policy.

Theoretically, we establish a data-dependent upper bound on the suboptimality of SAFARI for MF-MARL without the stringent assumptions on the sufficient coverage of the experience data (e.g., finite concentrability coefficients [11] or uniformly lower bounded densities of visitation measure [12]). More specifically, we only assume that the experience data of N training trajectories contains enough information about the optimal policy. Then we prove that for an episodic MF-MARL problem with a horizon H, SAFARI attains a sub-optimality gap of $\mathcal{O}(H^2 d_{\text{eff}}/\sqrt{N})$, where d_{eff} is the effective dimension of the function class (RKHS) for parameterizing the value function and independent on the number of agents.

In addition to the offline settings, our SAFARI algorithm can also be extended to MF-MARL under the online setting, which is of independent interest. Specifically, we only need to flip the sign of our uncertainty quantifier, and convert it to a bonus function for promoting the exploration [13]. We then prove that such an optimistic variant attains a sample efficient regret bound, which is also independent on the number of agents under the online setting.

2 Related Work

• **Mean-Field MARL**. Existing literature has proposed various mean-field approximation approaches to model the population behavior of the agents for MARL with a large number, even infinitely many homogenous agents. [14] investigate a mean-field game with deterministic linear state transitions, and reformulate it as a mean-field MDP, where the mean-field state lies in finite-dimensional probability simplex. [4] propose a mean-field approximation approach over actions, which approximates the interaction between any given agent and the population by the interaction between the agent's action and the averaged actions of its neighboring agents. Such an averaging approach over the local actions, however, is only applicable when a sparse graph over agents is given, which requires extensive prior knowledge. [5] investigate a mean-field MDP from the perspective of mean-field control. As the mean-field state lies in a probability simplex and continuous in nature, they propose to discretize the joint state-action space such that conventional RL algorithms can be applied. [15] investigate a mean-field MDP motivated by permutation invariance. They require a central controller managing the actions of all the agents, and therefore is restricted to handling the curse of many agents from the exponential blowup of joint state space. More recently, [6] investigate a similar mean-field MDP, which allows agents to make their own local actions without resorting to a centralized controller. All these methods focus the online settings. In comparison, our proposed SAFARI algorithm and theory focus on the offline settings.

• **RL for Mean-Field Game.** Our work is also related to the literature that studies RL methods for mean-field games [16–19]. Such a game can be viewed as the infinite-agent limit of general-sum Markov game with homogeneous agents, and the aggregated effect of the other agents is also summarized as a mean-field state. In contrast to mean-field MARL, the solution concept of mean-field game is the Nash equilibrium, which corresponds to a pair of a local policy π^* of the representative agent and a mean-field state d^* satisfying the following two properties: (i) when the mean-field state is set to d^*, π^* is the optimal policy of the representative agent; and (ii) when all agents adopt π^*, the resulting mean-field state is d^*. Recently, there are many recent works developing RL methods for solving mean-field games. See, e.g., [20–30] and the references therein. Most of these methods adopts a double-loop structure, where the inner loop finds the optimal local policy given the current mean field state and the outer loop updates the mean-field states. Moreover, these works often assume the data distribution is well-explored with either a generative model [31] or bounded concentrability coefficients [32]. Our mean-field MARL problem is similar to the inner-loop problem of finding the optimal local policy in mean-field games. In contrast to these existing works, our algorithm and

theory can be applied to datasets that are possibly not well-explored. Moreover, as mean-field MARL and mean-field games are different models, our work is not directly comparable to these works.

• **Offline Single-Agent RL**. Our work is also closely related to the literature on offline single-agent RL, which often focuses on either policy evaluation or policy optimization. In particular, in policy evaluation, the goal is to estimate the value function of a target policy, whereas in policy optimization, we aims to learn the optimal policy, which can be achieved via estimating the optimal value function. For both these tasks, in the offline setting, due to the lack of continuing exploration [33], the distribution shift [10] is a fundamental challenge. That is, the trajectories in the dataset and those induced by the target policy or the optimal policy might have diverse distributions. Such a challenge is further exacerbated when function approximators are adopted to represent the desired value functions. To overcome such a challenge, most of the existing theoretical works imposes certain well-exploration assumptions on the dataset. Some of commonly made assumptions include uniformly lower bounded visitation measure of the behavior policy, uniformly upper bounded importance sampling ratio, and bounded concentrability coefficients. See, e.g., [34–48, 11, 49–54, 12, 55–63] and the references therein.

However, in practice, such assumptions on the dataset often fail to hold [64–67]. In light of this, there is a line of recent works that proposes various pessimism-based offline single-agent RL algorithms with empirical evidence or theoretical guarantees [68–74]. In particular, [71] propose a regularized variant of fitted Q-iteration [34–36], which is shown to attain the optimal policy within a restricted policy class without assuming the dataset is well-explored. Moreover, with an arbitrary dataset, [72–74] identify the critical role of pessimism in achieving offline sample efficiency. Among these works, our work is particularly related to [73], which develops a pessimistic variant of the value iteration algorithm with finite-dimensional linear function approximation. In comparison, our SAFARI algorithm extends such an algorithm to mean-field MARL and we propose to employ RKHS mean embedding for handling the difference between finite-agent empirical mean-field state and its infinite-agent counterpart. Moreover, our algorithm and analysis involve infinite-dimensional RKHS, which strictly generalizes those in [73].

Notation: Given a space $\mathcal{X}$, we denote $\mathcal{M}(\mathcal{X})$ as the collection of probability distributions supported on $\mathcal{X}$. Let $u, v, w \in \mathcal{H}$ be elements in a Hilbert space, we denote $\langle u, v\rangle$ as the inner product, and $u \otimes v$ as the outer product satisfying $(u \otimes v)w = u\langle v, w\rangle$. For a scalar a, we denote $\{a\}^+ = \max\{0, a\}$. We use $\mathcal{O}(\cdot)$ to hide absolute constants and log factors.

3 Mean-Field Multi-Agent RL

We consider a Multi-Agent Reinforcement Learning (MARL) problem with $m+1$ agents and time horizon H. For the i-th agent (also known as the Representative Agent (RA)), at step h, we denote $s_{i,h} \in \mathcal{S}$ and $a_{i,h} \in \mathcal{A}$ as its state and action, respectively. We assume $\mathcal{S}$ and $\mathcal{A}$ are compact.

Different from single agent RL problem, the transition kernel, reward function, and policy of an representative agent in MARL depend not only on its individual state, but the states of m other agents. Furthermore, we assume that the interaction of the representative agent to the other agents is permutation invariant, i.e., the influence of all the other agents is modeled using the empirical distribution of states $\widehat{d}_{s,h} = \frac{1}{m}\sum_{j\neq i}^{m} \delta_{s_{j,h}} \in \mathcal{M}(\mathcal{S})$. To this end, we define the transition kernel $p_h : \mathcal{S} \times \mathcal{M}(\mathcal{S}) \times \mathcal{A} \mapsto \mathcal{M}(\mathcal{S})$, the (deterministic) reward function $r_h : \mathcal{S} \times \mathcal{M}(\mathcal{S}) \times \mathcal{A} \mapsto \mathbb{R}$, and the policy $\pi_h : \mathcal{S} \times \mathcal{M}(\mathcal{S}) \mapsto \mathcal{M}(\mathcal{A})$ all depending on a "meta state" denoted as $\widehat{\omega}_h = (s_{i,h}, \widehat{d}_{s,h}) \in \mathcal{S} \times \mathcal{M}(\mathcal{S})$. For simplicity, we denote $\Omega = \mathcal{S} \times \mathcal{M}(\mathcal{S})$ as the meta state space.

Remark 1. The empirical distribution of states $\widehat{d}_{s,h}$ is naturally permutation invariant and evolves according to the transition kernel p_h and policy π_h. To see this, suppose each agent takes the same policy π_h at step h. Then at step $h+1$, the state $s_{h+1,j}$ of the j-th agent is sampled from the distribution $p_h(\cdot \mid s_{h,j} \times \widehat{d}_{s,h}, a_{h,j})$, where $a_{h,j}$ is determined by policy $\pi_h(\cdot \mid s_{h,j} \times \widehat{d}_{s,h})$. Collecting m states $d_{h+1,j}$ for $j \neq i$ induces the empirical distribution of states $\widehat{d}_{s,h+1}$.

We now define several important notions in MARL. Given a policy π, the value function $V_h^\pi : \Omega \mapsto \mathbb{R}$ at step $h \leq H$ for a representative agent is

$$V_h^\pi(\omega) = \mathbb{E}_\pi\left[\sum_{i=h}^{H} r_i(\omega_i, a_i) \,\middle|\, \omega_h = \omega\right], \tag{1}$$

where $\mathbb{E}_\pi$ denotes the expectation over the randomness in trajectories induced by policy π. The action-value function (Q-function) $Q_h^\pi : \Omega \times \mathcal{A} \mapsto \mathbb{R}$ is defined as

$$Q_h^\pi(\omega, a) = \mathbb{E}_\pi \left[\sum_{i=h}^{H} r_i(\omega_i, a_i) \,\middle|\, \omega_h = \omega, a_h = a \right].$$

By definition, V_h^π and Q_h^π are related via $V_h^\pi(\omega) = \int_{\mathcal{A}} Q_h^\pi(\omega, a)\pi(a|\omega)da \triangleq \langle Q_h^\pi, \pi \rangle_{\mathcal{A}}$. Next, we define the Bellman operator and conditional transition operator. At each step $h \leq H$, the Bellman operator denoted as $\mathbb{B}_h$ is

$$\begin{aligned}(\mathbb{B}_h g)(\omega, a) &= \mathbb{E}\left[r_h(\omega_h, a_h) + g(\omega_{h+1}) \mid \omega_h = \omega, a_h = a\right] \\ &= r_h(\omega, a) + (\mathbb{P}_h g)(\omega, a),\end{aligned} \tag{2}$$

where g is a function defined on Ω, and $\mathbb{P}_h$ is referred to as the conditional transition operator.

Mean-Field MARL As the number of agents goes to infinity, the empirical distribution of states $\widehat{d}_s$ converges to a (continuous) limit d_s. Then the mean-field MARL problem for a representative agent is defined as a tuple $(\Omega, \mathcal{A}, H, P, r)$, where Ω and $\mathcal{A}$ are the meta state space and action space, respectively, H is the horizon, $P = \{p_h\}_{h=1}^H : \Omega \times \mathcal{A} \mapsto \mathcal{M}(\mathcal{S})$ is the transition kernel, and $r = \{r_h\}_{h=1}^H$ is the reward function defined on $\Omega \times \mathcal{A}$. Following Remark 1, the transition of d_s is also determined by $P = \{p_h\}_{h=1}^H$.

To tackle the infinite-dimensional joint distribution of states, we embed the meta state-action space $\Omega \times \mathcal{A}$ into a reproducing kernel Hilbert space (RKHS). Specifically, denote $\Xi = \mathcal{S} \times \mathcal{S} \times \mathcal{A}$ and let $K : \Xi \times \Xi \mapsto \mathbb{R}$ be a symmetric positive kernel. The corresponding feature mapping of kernel k is denoted as ψ, which verifies $\langle \psi(\cdot), \psi(\cdot) \rangle = K(\cdot, \cdot)$ can be infinite dimensional. For any $(\omega, a) \in \Omega \times \mathcal{A}$, we define mean embedding as

$$\mu(\omega, a) = \mathbb{E}_{s' \sim d_s}[\psi(s, s', a)]. \tag{3}$$

Based on the embedding, we parameterize the reward r_h and Markov transition p_h as linear functionals of $\mu(\omega, a)$ in RKHS $\mathcal{H}_K$ induced by kernel K, i.e.,

$$r_h(\omega, a) = \langle \mu(\omega, a), \theta_h \rangle, \quad p_h(\omega' \mid \omega, a) = \left\langle \mu_(\omega, a), v_h(\omega') \right\rangle, \tag{4}$$

where θ_h, v_h are understood as "weights" and have bounded Hilbert norm (see Assumption 3). Such a parameterization encodes a rich family of functions, once the kernel is universal [15]. By the definition of Q-function and value function, we can show that the Bellman operator can also be parameterized in $\mathcal{H}_K$.

Proposition 1. Suppose the reward function r_h and the transition kernel p_h is parameterized in $\mathcal{H}_K$ by (4) for $h = 1, \ldots, H$. Then for any $g : \Omega \mapsto \mathbb{R}$, the Bellman operator $(\mathbb{B}_h g)$ and conditional transition operator $(\mathbb{P}_h g)$ defined in (2) can be written as

$$(\mathbb{B}_h g)(\omega, a) = \langle \mu(\omega, a), w_g \rangle, \quad (\mathbb{P}_h g)(\omega, a) = \langle \mu(\omega, a), w_g + \theta_h \rangle,$$

where w_g depends on the function g.

The proof is provided in Appendix C.1, which follows from pure algebraic manipulation. From the perspective of policy learning in mean-field MARL, Proposition 1 motivates us to estimate the Bellman operator $\mathbb{B}_h$ in $\mathcal{H}_K$, and then optimize the estimated Q-function to obtain a policy. We introduce the detailed learning procedure in Section 4 (Algorithm 1).

4 Offline Pessimistic Value Iteration

In this section, we introduce our dataset and learning algorithm. We collect multiple trajectories of a representative agent in a mean-field MARL problem. Here the mean-field state distribution d_s is prohibitive to trace. Instead, we only independently observe the states of a finite number of agents. Accordingly, the batched dataset $\mathcal{D}_{N,H}$ consists of N trajectories of length H, within which the n-th sequence is $\tau_n = \left\{(s_h^n \in \mathcal{S}^{m+1}, a_h^n \in \mathcal{A}, r_h^n \in \mathbb{R})\right\}_{h=1}^H$. Without loss of generality, we assume $s_{h,0}$ is the state of the representative agent, and the reward function is bounded by 1, i.e., $|r_h(\omega, a)| \leq 1$ for any $\omega \in \Omega, a \in \mathcal{A}$. The collected trajectories are generated by some unknown behavior policy.

Recall $\widehat{d}_{s_h^n} = \frac{1}{m}\sum_{j=1}^m s_{h,j}^n$ is the empirical state distribution induced by s_h^n. (We slightly alter the notation to emphasize the empirical distribution is generated by the collection of m states $s_{h,1:m}^n$,

while in the previous context, we use a general purpose notation $\widehat{d}_{s,h}$.) We denote $\widehat{\omega}_h^n = s_{h,0}^n \times \widehat{d}_{s_h^n}$, and compute the empirical mean embedding of $(\widehat{\omega}_h^n, a_h^n)$ as

$$\mu(\widehat{\omega}_h^n, a_h^n) = \mathbb{E}_{s' \sim \widehat{d}_{s_h^n}}[\psi(s_{h,0}^n, s', a_h^n)] = \frac{1}{m}\sum_{j=1}^m \psi(s_{h,0}^n, s_{h,j}^n, a_h^n).$$

Under mild conditions, the empirical mean embedding $\mu(\widehat{\omega}_h^n, a_h^n)$ concentrates around the infinite agent mean embedding $\mu(\omega_h^n, a_h^n)$ defined in (3), where ω_h^n is the infinite agent meta state. See a detailed error quantification in Lemma 3.

Pessimistic Value Iteration Our goal is to learn an optimal policy to be deployed for all the agents based on the experience data of the representative agent. The idea is to estimate the Q-function at each time step in the RKHS $\mathcal{H}_K$, and then optimize the Q-function to obtain an optimal policy. In more detail, at step $h \le H$, we estimate Bellman operator by optimizing the empirical mean squared Bellman error

$$(\widehat{\mathbb{B}}_h \widehat{V}_{h+1}) = \operatorname*{argmin}_{f(\cdot)=\langle\cdot,\alpha\rangle \in \mathcal{H}_K} \sum_{n=1}^N \Big(f(\mu(\widehat{\omega}_h^n, a_h^n)) - r_h^n - \widehat{V}_{h+1}(\widehat{\omega}_{h+1}^n)\Big)^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{5}$$

where $\lambda \ge 1$ controls the regularization strength, $\widehat{V}$ is the estimated value function, and $\|\cdot\|_{\mathcal{H}}$ denotes the Hilbert norm.

The solution to (5) can be written in a closed form. For notational simplicity, we define

$$K((\omega, a), \cdot) = \mathbb{E}_{s' \sim d_s}[K((s, s', a), \cdot)] \quad \text{with} \quad \omega = s \times d_s.$$

Then we denote the gram matrix $K_h \in \mathbb{R}^{N \times N}$ as

$$[K_h]_{\ell,\ell'} = K((\widehat{\omega}_h^\ell, a_h^\ell), (\widehat{\omega}_h^{\ell'}, a_h^{\ell'})) \triangleq \mathbb{E}_{s_1 \sim \widehat{d}_h^\ell, s_2 \sim \widehat{d}_h^{\ell'}} \langle \psi(s_{h,0}^\ell, s_1, a_h^\ell), \psi(s_{h,0}^{\ell'}, s_2, a_h^{\ell'})\rangle$$

for $\ell, \ell' = 1, \ldots, N$. Meanwhile, for any (ω, a), we denote feature vector $\phi_h(\omega, a) = \big[K((\widehat{\omega}_h^1, a_h^1), (\omega, a)), \ldots, K((\widehat{\omega}_h^N, a_h^N), (\omega, a))\big]^\top \in \mathbb{R}^N$. Then the estimated Bellman operator $\widehat{\mathbb{B}}_h \widehat{V}_{h+1}$ can be written as

$$\begin{aligned} &(\widehat{\mathbb{B}}_h \widehat{V}_{h+1})(\omega, a) = \phi_h(\omega, a)^\top \widehat{\alpha}_h \\ \text{with} \quad &\widehat{\alpha}_h = (K_h + \lambda I)^{-1}[r_h^1 + \widehat{V}_{h+1}(\widehat{\omega}_{h+1}^1), \ldots, r_h^N + \widehat{V}_{h+1}(\widehat{\omega}_{h+1}^N)]^\top, \end{aligned} \tag{6}$$

We summarize the proposed SAFARI algorithm in Algorithm 1.

Algorithm 1 Pessimistic Mean-Field Value Iteration (SAFARI)

Input: Dataset $\mathcal{D}_{N,H}$, coefficient β, regularization coefficient λ.
Initialize: Set $\widehat{V}_{H+1} = 0$.
for $h = H, H-1, \ldots, 1$ **do**
  Compute $\Lambda_h = K_h + \lambda I$.
  Estimate $\widetilde{Q}_h(\omega, a) \triangleq (\widehat{\mathbb{B}}_h \widehat{V}_{h+1})(\omega, a) = \phi_h(\omega, a)^\top \widehat{\alpha}_h$ as in (6).
  Set $\Gamma_h(\omega, a) = \beta \cdot \lambda^{-1/2} \big(K((\omega, a), (\omega, a)) - \phi_h(\omega, a)^\top \Lambda_h^{-1} \phi_h(\omega, a)\big)^{1/2}$.
  Let $\widehat{Q}_h(\omega, a) = \min\{\widetilde{Q}_h(\omega, a) - \Gamma_h(\omega, a), H - h + 1\}^+$.
  Optimal policy $\widehat{\pi}_h = \operatorname{argmax}_\pi \langle \widehat{Q}_h(\omega, \cdot), \pi(\cdot \mid \omega)\rangle_{\mathcal{A}}$.
  Set $\widehat{V}_h(\omega) = \langle \widehat{Q}_h(\omega, \cdot), \widehat{\pi}_h(\cdot \mid \omega)\rangle_{\mathcal{A}}$.
end for
Output: Estimated Q-function $\widehat{Q}_h$, value function $\widehat{V}_h$, and optimal policy $\widehat{\pi}_h$ for $h = 1, \ldots, H$.

The quantity Γ_h quantifies the uncertainty in estimating the Bellman operator $\mathbb{B}_h \widehat{V}_{h+1}$ using kernel ridge regression. We subtract Γ_h for estimating the Bellman operator to account for the spurious correlation in the experience data (see Technical Overview following Theorem 1 for a detailed explanation). We truncate $\widehat{Q}_h$ at $H - h + 1$, since the reward function is bounded by 1.

5 Suboptimality of Policy Learned by SAFARI

We investigate the performance of the optimal policy $\widehat{\pi}$ learned by Algorithm 1. Before we proceed, we state the following assumptions.

Assumption 1 (Boundedness of Kernel)**.** Kernel $K(\cdot,\cdot)$ is bounded, i.e., without loss of generality, we assume $\sup_{\xi\in\Xi}|K(\xi,\xi)|\leq 1$.

By Cauchy-Schwarz inequality, Assumption 1 implies for any $\xi_1,\xi_2\in\Xi$, $K(\xi_1,\xi_2)\leq\sqrt{K(\xi_1,\xi_1)K(\xi_2,\xi_2)}\leq 1$. Such an assumption holds for a rich family of commonly used kernels, e.g., RBF kernel and Laplacian kernel, and is a standard assumption in literature [75, 76].

The second assumption characterizes the spectrum of kernel K. We first introduce the integral operator induced by kernel K. Let $f:\Xi\mapsto\mathbb{R}$ be a square-integrable function. Then we define the integral operator $\mathcal{T}_K$ as

$$(\mathcal{T}_K f)(\xi)=\int K(\xi,x)f(x)dx \quad \text{for} \quad \xi\in\Xi.$$

By Mercer's theorem [77], $\mathcal{T}_K$ has corresponding positive eigenvalues σ_i and eigenfunctions ν_i. Then the kernel K admits a decomposition

$$K(\xi_1,\xi_2)=\sum_{i=1}^{\infty}\sigma_i\nu_i(\xi_1)\nu_i(\xi_2).$$

Assumption 2 (Spectrum of Kernel)**.** The eigenvalue σ_i satisfies one of the following three conditions:

1. *(Finite Spectrum)*. There exists a positive integer γ, such that $\sigma_i=0$ for all $i>\gamma$.
2. *(Exponential Decay)*. There exist positive constants C_1,C_2 and exponent $\gamma>0$ such that $\sigma_i\leq C_1\exp(-C_2 i^{\gamma})$.
3. *(Polynomial Decay)*. There exists a positive constant C and exponent $\gamma\geq 3+\mathcal{O}(\frac{1}{d})$ such that $\sigma_i\leq Ci^{-\gamma}$, where d is the dimension of $\mathcal{S}\times\mathcal{S}\times\mathcal{A}$.

Furthermore, in *(Exponential Decay)* and *(Polynomial Decay)*, we assume the eigenfunction ν_i is uniformly bounded, i.e., $\sup_i\|\nu_i\|_\infty\leq 1$.

As we will show in our theory, the decay rate of the spectrum significantly influences the performance of the proposed SAFARI algorithm. We give examples to better interpret the three categories above. In *(Finite Spectrum)* case, by (4), the reward function and transition kernel is a linear function of a finite dimensional feature map. Such a parameterization is satisfied by linear MDP [78, 79]. In *(Exponential Decay)* and *(Polynomial Decay)* cases, the feature map is infinite dimensional. For example, RBF kernel belongs to *(Exponential Decay)* case, while Laplacian kernel and neural tangent kernel belong to *(Polynomial Decay)* case. We assume $\gamma\geq 3+\mathcal{O}(1/d)$ in *(Polynomial Decay)* for technical simplicity, yet it is not restrictive: Laplacian kernel and neural tangent kernel both have a polynomial decay rate of $\gamma=d$ [80].

The last assumption imposes some regularity on the reward function and transition probabilities.

Assumption 3 (Boundedness)**.** The weights θ_h and v_h in reward function r_h and Markov transition kernel p_h are bounded for any $h=1,\ldots,H$, respectively, i.e., $\|\theta_h\|_{\mathcal{H}}\leq 1$ and $\int_\Omega\|v_h(x)\|_{\mathcal{H}}\,dx\leq\sqrt{d_{\text{eff}}}$, where $d_{\text{eff}}=\sup_{K_h}\log\det(I+K_h/\lambda)$ is the effective dimension of $\mathcal{H}_K$ with supremum over all Gram matrix $K_h\in\mathbb{R}^{N\times N}$.

The effective dimension describes the complexity of $\mathcal{H}_K$ for parameterizing the MDP [81], whose scale is closely related to the spectrum of kernel K. In the special case of K having a γ-finite spectrum as in Assumption 2, we have $d_{\text{eff}}=\mathcal{O}(\gamma)$, which resembles the dimensionality of a finite dimensional Euclidean space.

We measure the pointwise suboptimality of the learned policy $\widehat{\pi}$. We define the global optimal policy by the recursion,

$$\pi_h^*=\operatorname*{argmax}_{\pi}\ \langle Q_h^*,\pi\rangle_{\mathcal{A}}\,,\quad \text{with}\quad Q_h^*=\mathbb{B}_h V_{h+1}^*, V_h^*=\langle Q_h^*,\pi_h^*\rangle_{\mathcal{A}}\,,\ \text{and}\ V_{H+1}^*=0.$$

Then the suboptimality of $\widehat{\pi}$ is given as

$$\texttt{SubOpt}(\widehat{\pi};\omega) = V_1^{\pi^*}(\omega) - V_1^{\widehat{\pi}}(\omega).$$

Our main result is provided in the following theorem, which upper bounds $\texttt{SubOpt}(\widehat{\pi};\omega)$.

Theorem 1. Suppose Assumption 1 – 3 hold. For any $\delta \in (0,1)$, let $\widehat{\pi}_h$ be the policy returned by Algorithm 1 with

$$m \geq \log(2/\delta), \quad \lambda = 1, \quad \beta = \begin{cases} c\max\{d,\gamma\}H\sqrt{\log(\max\{d,\gamma\}HN/\delta)} & \textit{(Finite Spectrum)} \\ cH\sqrt{d\left(\log(HN/\delta)\right)^{1+2/\gamma}} & \textit{(Exponential Decay)} \\ cN^{\frac{d+1}{d+\gamma}}H\sqrt{d\log(HN/\delta)} & \textit{(Polynomial Decay)} \end{cases},$$

where d is the dimension of $\Xi = \mathcal{S}\times\mathcal{S}\times\mathcal{A}$ and c is some constant depending on C, C_1, C_2 and Lebesgue measure of Ξ. Then for any meta state ω, with probability at least $1-\delta$ over the randomness of the dataset $\mathcal{D}_{N,H}$, we have

$$\texttt{SubOpt}(\widehat{\pi};\omega) \leq 2\sum_{h=1}^{H}\mathbb{E}_{\pi^*}[\Gamma_h(\omega_h,a_h) \mid \omega_1 = \omega].$$

Theorem 1 indicates that the suboptimality of learned policy depends on the uncertainty quantifier Γ_h. The scale of Γ_h depends on how well the collected data explore the state-action space. Moreover, from a Bayesian learning perspective, Γ_h measures the eliminated uncertainty in estimating the Bellman operator given dataset $\mathcal{D}_{N,H}$ [73]. To better understand the convergence of SubOpt, we specialize Theorem 1 under a weak data coverage assumption.

Assumption 4 (Weak Coverage)**.** Suppose the dataset is collected under some behavior policy $\bar{\pi}$ such that there exists a constant $c_{\min} > 0$ satisfying

$$\inf_{\|f\|_{\mathcal{H}}=1}\langle f, \mathbb{E}_{\bar{\pi}}[\mu(\omega_h,a_h)\otimes\mu(\omega_h,a_h)]f\rangle \geq c_{\min} \quad \text{for any} \quad h = 1,\ldots,H.$$

Recall that μ is the mean embedding in $\mathcal{H}_K$.

Assumption 4 says that the operator $\mathbb{E}_{\bar{\pi}}[\mu(\omega_h,a_h)\otimes\mu(\omega_h,a_h)]$ is positive definite. Intuitively, this requires that the collected data relatively well spread over the state-action space. We present the following Corollary providing a concrete convergence rate of SubOpt.

Corollary 1. Under the setting in Theorem 1, we additionally assume Assumption 4 holds. Then for $N \geq \Omega(\log(d_{\text{eff}}H/\delta))$ sufficiently large, with probability $1-\delta$, we have

$$\texttt{SubOpt}(\widehat{\pi};\omega) = \mathcal{O}\left(H^2 d_{\text{eff}}\sqrt{\frac{\log(d_{\text{eff}}HN/\delta)}{N}}\right).$$

Here d_{eff} is the effective dimension of RKHS $\mathcal{H}_K$, which takes value

$$d_{\text{eff}} = \begin{cases} \max\{d,\gamma\}\log N & \textit{(Finite Spectrum)} \\ d(\log N)^{1+1/\gamma} & \textit{(Exponential Decay)} \\ dN^{\frac{d+1}{d+\gamma}}\log N & \textit{(Polynomial Decay)} \end{cases}.$$

Impact of Kernel Spectrum The spectrum of kernel K significantly influences the performance of the learned policy. In *(Finite Spectrum)* case, the effective dimension scales linearly with dimension d and γ, and SubOpt converges at a rate of $\mathcal{O}(H^2\max\{d,\gamma\}/\sqrt{N})$, which recovers the result of Corollary 4.5 in [73] on linear MDP. In *(Exponential Decay)* case, the convergence rate is $\mathcal{O}(H^2 d(\log N)^{1+1/\gamma}/\sqrt{N})$, which is similar to *(Finite Spectrum)* case with additional logarithmic dependence on N. However, in *(Polynomial Decay)* case, the convergence rate is considerably slower, and relies heavily on the decay rate γ. Consider, for instance, Laplacian kernel and NTK, whose spectrum decays with $\gamma = d$. Then SubOpt converges at a rate of $\mathcal{O}(H^2 dN^{-\frac{1}{2d}}\log N)$, which suffers from the curse of dimensionality without further assumptions on data.

No Curse of Many Agents The convergence of SubOpt does not suffer from the curse of many agents. In particular, both Theorem 1 and Corollary 1 only impose a mild requirement on the number m of neighboring agents to be sampled. This is due to the permutation invariance in mean-field MARL, since the interactive influence of neighboring agents are captured by the distribution of states.

Technical Overview We briefly discuss the proof of Theorem 1 and Corollary 1. The full proof is deferred to Appendix A and B. We first decompose $\texttt{SubOpt}$ into three terms (see Lemma 1):

$$\texttt{SubOpt}(\pi;\omega) = \mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3.$$

Here $\mathcal{E}_1 = \sum_{h=1}^H \mathbb{E}_{\widehat{\pi}}[\widehat{Q}_h(\omega_h, a_h) - (\mathbb{B}_h\widehat{V}_{h+1})(\omega_h, a_h) \mid \omega_1 = \omega]$ reflects the uncertainty in estimating the Bellman operator. Note that the evaluating trajectory is generated by the learned policy $\widehat{\pi}$, which has spurious correlation with the estimated Bellman operator; $\mathcal{E}_2 = \sum_{h=1}^H \mathbb{E}_{\pi^*}[(\mathbb{B}_h\widehat{V}_{h+1})(\omega_h, a_h) - \widehat{Q}_h(\omega_h, a_h) \mid \omega_1 = \omega]$ is the estimation error of Bellman operator again, yet it is evaluated by a trajectory generated by π^*. Compared to $\mathcal{E}_1$, $\mathcal{E}_2$ does not suffer from the spurious correlation between the learned policy and the estimated Bellman operator. Lastly, $\mathcal{E}_3 = \sum_{h=1}^H \mathbb{E}_{\pi}[\langle\widehat{Q}_h(\omega_h, \cdot), \pi_h^*(\cdot \mid \omega_h) - \widehat{\pi}_h(\cdot \mid \omega_h)\rangle_{\mathcal{A}} \mid \omega_1 = \omega]$ is the optimization error. By the optimality of $\widehat{\pi}$, we immediately have $\mathcal{E}_3 \le 0$.

In order to tackle $\mathcal{E}_1$ and $\mathcal{E}_2$, we properly choose Γ_h so that the event $E = \{|\mathbb{B}_h\widehat{V}_{h+1} - \widehat{\mathbb{B}}_h\widehat{V}_{h+1}| \le \Gamma_h\}$ happens with high probability. In fact, Γ_h is understood as the uncertainty quantifier of estimating $\mathbb{B}_h\widehat{V}_{h+1}$ with high confidence $1-\delta$. Then we can show $\mathcal{E}_1 \le 0$ conditioned on event E, meanwhile $\mathcal{E}_2 \le 2\sum_{h=1}^H \mathbb{E}_{\pi^*}[\Gamma_h(\omega_h, a_h) \mid \omega_1 = \omega]$. To this end, we reduce the upper bound of $\texttt{SubOpt}$ to bounding the uncertainty quantifier Γ_h, which allows us to leverage statistical tools. In particular, Γ_h consists of two types of statistical error: 1) covariate concentration error on mean embedding, i.e., finite agent empirical embedding $\mu(\widehat{\omega}_h^n, a_h^n)$ concentration error with respect to population counterpart $\mu(\omega_h^n, a_h^n)$; 2) regression error in Bellman operator estimation. We bound 1) by concentration of empirical means in Hilbert spaces (see Lemma 3). In bounding 2), we exploit the closed form solution of kernel ridge regression and concentration of self-normalizing processes (see Lemma 5).

6 Online Policy Learning

We extend our pessimistic policy learning to online mean-field settings. Different from performing backward value iteration, online mean-field MARL aims to learn a policy forward in time. Specifically, we consider the mean-field MARL problem $(\Omega, \mathcal{A}, H, P, r)$ again, where we denote s_0 as the representative agent and sample m neighboring agents from the mean-field distribution of states. We simulate N episodes of length H. In the n-th episode, we sample a starting meta state ω_1^n from a fixed initial distribution on Ω. However, we can only observe the states of m agents to approximate the mean-field state distribution, and denote $\widehat{\omega}_1^n$ as the received empirical counterpart of ω_1^n. Then the representative agent determines a policy $\pi^n = \{\pi_h^n\}_{h=1}^H$, to be deployed for all the agents. At step h, the representative agent takes an action a_h^n sampled from $\pi_h^n(\cdot \mid \widehat{\omega}_h^n)$, and transits to the next state following the Markov transition p_h. The reward function r_h^n is revealed to the representative agent after taking the action. Note that the reward function can be chosen by the environment adversarially. The n-th episode ends, when the representative agent receives the terminal reward $r_H^n(\widehat{\omega}_H^n, a_H^k)$. We define regret of the learned policy, competing with the globally optimal policy:

$$\texttt{Regret}(\{\pi^n\}_{n=1}^N; H) = \sup_{\pi} \sum_{n=1}^N \left(V_1^{\pi,n}(\omega_1^n) - V_1^{\pi^n,n}(\omega_1^n)\right),$$

where the value function V is defined in (1).

Similar to the offline setting, we parameterize the transition kernel $P = \{p_h\}_{h=1}^H$ in an RKHS. Specifically, we slightly abuse the notation to overload $\Xi = \mathcal{S} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{S}$. We consider a kernel $K : \Xi \times \Xi \mapsto \mathbb{R}$ with the corresponding feature map ψ. Then we parameterize the transition kernel p_h as

$$p_h(\omega' \mid \omega, a) = \langle\mu(\omega, a, \omega'), v_h\rangle \quad \text{with} \quad \mu(\omega, a, \omega') = \mathbb{E}_{s\sim d_s, s'\sim d_s'}[\psi(s_0, s, a, s_0', s')],$$

where v_h is the weight and we denote $\omega = (s_0, d_s), \omega' = (s_0', d_s')$. Here μ is the mean embedding depending additionally on the next meta state. We also note the slight variant of parameterizing the transition kernel compared to the off-line setting in (4). Furthermore, we assume the received reward $r_h^n(\omega, a) = \mathbb{E}_{s\sim d_s}[g_h^n(s_0, s, a)]$, where g_h^n represents the pairwise reward between the representative agent and its neighbors. In the sequel, we assume

$$\|v_h\|_{\mathcal{H}} \le 1 \quad \text{and} \quad \|g_h^n\|_\infty \le 1 \quad \text{for} \quad h = 1, \ldots, H,\ n = 1, \ldots, N.$$

We propose Online Mean-field Proximal Policy Optimization (OMPPO) in Algorithm 2 for solving the online mean-field MARL problem.

Algorithm 2 Online Mean-field Proximal Policy Optimization (OMPPO)

Input: Coefficient β, constant α.
Initialize: Set $\{\pi_0^n\}_{h=1}^H$ to be uniform on $\mathcal{A}$; $Q_h^0 = 0$ for $h = 1, \ldots, H$.
for $n = 1, \ldots, N$ **do**
  Sample starting state $\widehat{\omega}_1^n$.
  for $h = 1, \ldots, H$ **do**
    Set $\pi_h^n(a \mid \omega) \propto \pi_h^{n-1}(a \mid \omega) \exp(\alpha Q_h^{n-1}(\omega, a))$.
    Take action $a_h^n \sim \pi_h^n(\cdot \mid \widehat{\omega}_h^n)$; receive reward function r_h^n; transit to next state $\widehat{\omega}_{h+1}^n$.
  end for
  Set $V_{H+1}^n = 0$.
  for $h = H, H-1, \ldots, 1$ **do**
    Compute Gram matrix K_h^{n-1} and vector $\phi_h^n(\omega, a)$: For $\ell, \ell' = 1, \ldots, n-1$

$$[K_h^{n-1}]_{\ell,\ell'} = \int_{\Omega\times\Omega} K((\widehat{\omega}_h^\ell, a_h^\ell, \omega'), (\widehat{\omega}_h^{\ell'}, a_h^{\ell'}, \omega''))V_{h+1}^\ell(\omega')V_{h+1}^{\ell'}(\omega'')d\omega' d\omega''$$

$$[\phi_h^n(\omega, a)]_\ell = \int_{\Omega\times\Omega} K((\widehat{\omega}_h^\ell, a_h^\ell, \omega'), (\omega, a, \omega''))V_{h+1}^\ell(\omega')V_{h+1}^n(\omega'')d\omega' d\omega''.$$

    Compute $\Lambda_h^{n-1} = K_h^{n-1} + \lambda I$.
    Estimate $\widetilde{Q}_h^n(\omega, a) = \phi_h^{n-1}(\omega, a)^\top (\Lambda_h^{n-1})^{-1}[V_{h+1}^1(\widehat{\omega}_{h+1}^1), \ldots, V_{h+1}^{n-1}(\widehat{\omega}_{h+1}^{n-1})]^\top$.
    Set $\Gamma_h^n(\omega, a) = \beta \cdot \lambda^{-1/2} \left(K((\omega, a), (\omega, a)) - \phi_h^{n-1}(\omega, a)^\top (\Lambda_h^{n-1})^{-1}\phi_h^{n-1}(\omega, a)\right)^{1/2}$.
    Let $Q_h^n(\omega, a) = \min\left\{\widetilde{Q}_h^n(\omega, a) + r_h^n(\omega, a) + \Gamma_h^n(\omega, a), H - h + 1\right\}^+$.
    Compute $V_h^n(\omega) = \langle Q_h^n(\omega, \cdot), \pi_h^n(\cdot \mid \omega)\rangle_{\mathcal{A}}$.
  end for
end for
Output: Value function V_h^n and optimal policy π_h^n for $h = 1, \ldots, H$ and $n = 1, \ldots, N$.

Note that different from the off-line setting, OMPPO iteratively improves its policy while accumulating experience data. During the $(n-1)$-th episode, OMPPO executes the policy π^{n-1} and collects the trajectory data of the episode. At the end of the $(n-1)$-th episode, OMPPO uses the collected samples to update the estimated Q-functions $\{Q_h^{n-1}\}_{h=1}^H$, and obtains the improved policy π^n using the updated Q-functions. Further, we flip the sign of Γ_h to promote exploration.

Recall that d_{eff} is the effective dimension of $\mathcal{H}_K$ and takes values as in Corollary 1 based on the spectrum of K. We then present the following regret bound.

Theorem 2. Suppose Assumption 1 and 2 hold. Given $\delta \in (0, 1)$, we further assume the cardinality of the action space is bounded by $\log|\mathcal{A}| = \mathcal{O}(d_{\text{eff}}^2 \log^2(d_{\text{eff}} HN/\delta))$. Let π^n be the policy learned by Algorithm 2 with $\alpha = \sqrt{2\log|\mathcal{A}|/(H^2N)}$, $m \geq 2H^4N^2\log(2HN/\delta)$, $\lambda = 1$, and

$$\beta = \begin{cases} c\max\{d, \gamma\}H\sqrt{\log(\max\{d, \gamma\}HN/\delta)} & (\textit{Finite Spectrum}) \\ cH\sqrt{d\left(\log(HN/\delta)\right)^{1+2/\gamma}} & (\textit{Exponential Decay}) \\ cN^{\frac{d+1}{d+\gamma}}H\sqrt{d\log(HN/\delta)} & (\textit{Polynomial Decay}) \end{cases},$$

where d is the dimension of Ξ and c is some constant depending on C, C_1, C_2, and Lebesgue measure of Ξ. Then with probability $1-\delta$, we have

$$\texttt{Regret}(\{\pi^n\}_{n=1}^N; H) = \mathcal{O}\left(H^2 d_{\text{eff}}\sqrt{N}\log(HN/\delta)\right).$$

The regret in Theorem 2 does not suffer from the curse of many agents, thanks to the permutation invariance in mean-field MARL. We also observe that the spectrum of kernel has significant influence on the regret of learned policy. The proof is provided in Appendix D.

7 Conclusion

This paper proposes a SAFARI (Pessimistic Mean-Field Value Iteration) algorithm in mean-field MARL, with an extension to online settings (OMPPO algorithm). We prove a suboptimality bound $\mathcal{O}(H^2 d_{\text{eff}}/\sqrt{N})$ in the off-line setting, under a weak data coverage assumption. The suboptimality bound is free of the curse of many agents due to the permutation invariance in mean-field formulation. In online settings, we provide a regret bound of the order $\mathcal{O}(H^2 d_{\text{eff}}\sqrt{N})$.

References

[1] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037*, 2017.

[2] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[3] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[4] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5571–5580. PMLR, 2018.

[5] René Carmona, Mathieu Laurière, and Zongjun Tan. Model-free mean-field reinforcement learning: mean-field mdp and mean-field q-learning. *arXiv preprint arXiv:1910.12802*, 2019.

[6] Yan Li, Lingxiao Wang, Jiachen Yang, Ethan Wang, Zhaoran Wang, Tuo Zhao, and Hongyuan Zha. Permutation invariant policy optimization for mean-field multi-agent reinforcement learning: A principled approach. *arXiv preprint arXiv:2105.08268*, 2021.

[7] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

[8] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pages 1077–1084, 2014.

[9] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

[10] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[11] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

[12] Ming Yin, Yu Bai, and Yu-Xiang Wang. Near optimal provable uniform convergence in off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:2007.03760*, 2020.

[13] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

[14] Jiachen Yang, Xiaojing Ye, Rakshit Trivedi, Huan Xu, and Hongyuan Zha. Learning deep mean field games for modeling large population behavior. *arXiv preprint arXiv:1711.03156*, 2017.

[15] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Breaking the curse of many agents: Provable mean embedding q-iteration for mean-field reinforcement learning. In *International Conference on Machine Learning*, pages 10092–10103. PMLR, 2020.

[16] Minyi Huang, Peter E Caines, and Roland P Malhamé. Individual and mass behaviour in large population stochastic wireless power control problems: centralized and nash equilibrium solutions. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 1, pages 98–103. IEEE, 2003.

[17] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. i–le cas stationnaire. *Comptes Rendus Mathématique*, 343(9):619–625, 2006.

[18] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. ii–Horizon fini et contrôle optimal. *Comptes Rendus Mathématique*, 343(10):679–684, 2006.

[19] Minyi Huang, Peter E Caines, and Roland P Malhamé. Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized $varepsilon$-Nash equilibria. *IEEE transactions on automatic control*, 52(9):1560–1571, 2007.

[20] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. *arXiv preprint arXiv:1901.09585*, 2019.

[21] Xin Guo, Renyuan Xu, and Thaleia Zariphopoulou. Entropy regularization for mean field games with learning. *arXiv preprint arXiv:2010.00145*, 2020.

[22] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. A general framework for learning mean-field games. *arXiv preprint arXiv:2003.06069*, 2020.

[23] Zuyue Fu, Zhuoran Yang, Yongxin Chen, and Zhaoran Wang. Actor-critic provably finds nash equilibria of linear-quadratic mean-field games. In *International Conference on Learning Representations*, 2019.

[24] Berkay Anahtarcı, Can Deha Karıksız, and Naci Saldi. Fitted q-learning in mean-field games. *arXiv preprint arXiv:1912.13309*, 2019.

[25] Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Q-learning in regularized mean-field games. *arXiv preprint arXiv:2003.12151*, 2020.

[26] Berkay Anahtarcı, Can Deha Karıksız, and Naci Saldi. Value iteration algorithm for mean-field games. *Systems & Control Letters*, 143:104744, 2020.

[27] Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. *arXiv preprint arXiv:2007.03458*, 2020.

[28] Romuald Elie, Julien Pérolat, Mathieu Laurière, Matthieu Geist, and Olivier Pietquin. On the convergence of model free learning in mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7143–7150, 2020.

[29] Muhammad Aneeq uz Zaman, Kaiqing Zhang, Erik Miehling, and Tamer Başar. Reinforcement learning in non-stationary discrete-time linear-quadratic mean-field games. In *IEEE Conference on Decision and Control (CDC)*, pages 2278–2284. IEEE, 2020.

[30] Kai Cui and Heinz Koeppl. Approximately solving mean field games via entropy-regularized deep reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1909–1917. PMLR, 2021.

[31] Mohammad Gheshlaghi Azar, Rémi Munos, and Bert Kappen. On the sample complexity of reinforcement learning with a generative model. *arXiv preprint arXiv:1206.6461*, 2012.

[32] Rémi Munos. Performance bounds in ℓ_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007.

[33] Csaba Szepesvári. *Algorithms for reinforcement learning*. Morgan & Claypool, 2010.

[34] András Antos, Csaba Szepesvári, and Rémi Munos. Fitted Q-iteration in continuous action-space MDPs. In *Advances in Neural Information Processing Systems*, 2007.

[35] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

[36] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

[37] Amir-massoud Farahmand, Csaba Szepesvári, and Rémi Munos. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*, 2010.

[38] Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized policy iteration with nonparametric function spaces. *Journal of Machine Learning Research*, 17(1):4809–4874, 2016.

[39] Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of Tetris. *Journal of Machine Learning Research*, 16:1629–1676, 2015.

[40] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661, 2016.

[41] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

[42] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456, 2018.

[43] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, 2018.

[44] Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pages 9668–9678, 2019.

[45] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, 2019.

[46] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[47] Ziyang Tang, Yihao Feng, Lihong Li, Dengyong Zhou, and Qiang Liu. Doubly robust bias reduction in infinite horizon off-policy estimation. *arXiv preprint arXiv:1910.07186*, 2019.

[48] Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. GenDICE: Generalized offline estimation of stationary values. In *International Conference on Learning Representations*, 2020.

[49] Nathan Kallus and Masatoshi Uehara. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *arXiv preprint arXiv:1909.05850*, 2019.

[50] Nathan Kallus and Masatoshi Uehara. Doubly robust off-policy value and gradient estimation for deterministic policies. *arXiv preprint arXiv:2006.03900*, 2020.

[51] Nan Jiang and Jiawei Huang. Minimax value interval for off-policy evaluation and policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

[52] Masatoshi Uehara, Jiawei Huang, and Nan Jiang. Minimax weight and Q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, pages 9659–9668, 2020.

[53] Yaqi Duan, Zeyu Jia, and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning*, pages 2701–2709, 2020.

[54] Ming Yin and Yu-Xiang Wang. Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958, 2020.

[55] Ofir Nachum and Bo Dai. Reinforcement learning via Fenchel-Rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

[56] Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized Lagrangian. In *Advances in Neural Information Processing Systems*, 2020.

[57] Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. *arXiv preprint arXiv:2008.00483*, 2020.

[58] Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep Q-learning. In *Learning for Dynamics and Control*, pages 486–489, 2020.

[59] Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*, 2020.

[60] Tengyang Xie and Nan Jiang. $Q^\star$-approximation schemes for batch reinforcement learning: A theoretical comparison. *arXiv preprint arXiv:2003.03924*, 2020.

[61] Peng Liao, Zhengling Qi, and Susan Murphy. Batch policy learning in average reward Markov decision processes. *arXiv preprint arXiv:2007.11771*, 2020.

[62] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvári, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, 2020.

[63] Tongzheng Ren, Jialian Li, Bo Dai, Simon S Du, and Sujay Sanghavi. Nearly horizon-free offline reinforcement learning. *arXiv preprint arXiv:2103.14077*, 2021.

[64] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

[65] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114, 2020.

[66] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[67] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Tom Le Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, et al. RL Unplugged: Benchmarks for offline reinforcement learning. *arXiv preprint arXiv:2006.13888*, 2020.

[68] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

[69] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

[70] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

[71] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.

[72] Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. *arXiv preprint arXiv:2009.06799*, 2020.

[73] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? *arXiv preprint arXiv:2012.15085*, 2020.

[74] Chenjun Xiao, Yifan Wu, Tor Lattimore, Bo Dai, Jincheng Mei, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. On the optimality of batch policy optimization algorithms. *arXiv preprint arXiv:2104.02293*, 2021.

[75] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[76] Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. *arXiv preprint arXiv:1202.6504*, 2012.

[77] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[78] Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019.

[79] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

[80] Alberto Bietti and Francis Bach. Deep equals shallow for relu networks in kernel regimes. *arXiv preprint arXiv:2009.14397*, 2020.

[81] Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I. Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces, 2020.

[82] Yasemin Altun and Alex Smola. Unifying divergence minimization and statistical inference via convex duality. In *International Conference on Computational Learning Theory*, pages 139–153. Springer, 2006.

[83] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

[84] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No] *We do not foresee direct negative societal impacts of this work.*
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] *See Assumption 1 – 4 in Section 5.*
   (b) Did you include complete proofs of all theoretical results? [Yes] *The full proofs of theoretical results are provided in the supplementary, organized in sections corresponding to the main text of the paper.*
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] *This submisstion does not provide empirical results.*
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]