A Single Online Agent Can Efficiently Learn Mean Field Games

Anonymous Author
Anonymous Institution

Abstract

Mean field games (MFGs) are a promising framework for modeling the behavior of large-population systems. However, solving MFGs can be challenging due to the coupling of forward population evolution and backward agent dynamics. Typically, obtaining mean field Nash equilibria (MFNE) involves an iterative approach where the forward and backward processes are solved alternately, known as fixed-point iteration (FPI). This method requires fully observed population propagation and agent dynamics over the entire spatial domain, which could be impractical in some real-world scenarios. To overcome this limitation, this paper introduces a novel online single-agent model-free learning scheme, which enables a single agent to learn MFNE using online samples, without prior knowledge of the state-action space, reward function, or transition dynamics. Specifically, the agent updates its policy through the value function (Q), while simultaneously evaluating the mean field state (M), using the same batch of observations. We develop two variants of this learning scheme: off-policy and on-policy QM iteration. We prove that they efficiently approximate FPI, and a sample complexity guarantee is provided. The efficacy of our methods is confirmed by numerical experiments.

1 Introduction

Mean field games (MFGs) (Huang et al., 2006; Lasry and Lions, 2007) offer a tractable model to describe the population impact on individual agents in multi-agent systems with a large population. This work delves into the increasingly prominent field of applying reinforcement learning (RL) (Sutton and Barto, 2018) to learn MFGs.

In an MFG, the influence of other agents is encapsulated by a *population mass* which provides a reliable approximation of real interactions between agents when the number of agents is large. A widely used method for learning MFGs is fixed-point iteration (FPI), which iteratively calculates the *best response* (BR) w.r.t. the current population, and the *induced population distribution* (IP) w.r.t. the current policy (Guo et al., 2019). The FPI algorithm can be formally expressed as:

$$(\pi_k, \mu_k) = (\Gamma_{\mathrm{IP}} \circ \Gamma_{\mathrm{BR}})^k (\pi_0, \mu_0),$$

where operators Γ_{BR} calculates the best response and Γ_{IP} calculates the induced population distribution. We defer the full definitions of these operators to Section 2.

Although it is a prominent scheme for learning MFGs, current implementations of FPI and its variants face several limitations, especially in the IP calculation: 1) Γ_{BR} and Γ_{IP} are implemented separately and executed alternately, impeding parallel computing and potentially increasing the *computational complexity* of the entire algorithm. 2) The implementation of Γ_{IP} typically requires the knowledge of the transition dynamics of the environment (Yang et al., 2017; Perrin et al., 2021; Chen et al., 2023a,b), limiting the use of *model-free* methods. 3) Despite some proposals of model-free strategies in existing literature, these methods demand direct observability of population dynamics (Carmona et al., 2019; Lee et al., 2021; Anahtarci et al., 2023). In reality, fulfilling this requirement generally needs a central server capable of communication across the entire state space, restricting the feasibility of implementing such methods with a single online agent, i.e., an agent that interacts with the environment and collects local observations to learn and act on-the-go.

While these limitations paint part of the picture, we still need to answer the following question:

> *Why should we employ a single online agent to learn the equilibria of mean field games?*

The reasons are multifold:

Preliminary work. Under review by AISTATS 2024. Do not distribute.

- In many real-world scenarios, a single online agent is often the most accessible, and sometimes the only available resource (Shou et al., 2022).
- Online single-agent model-free methods are more straightforward to implement, since they do not require prior knowledge of the data or the model.
- Once a single-agent model-free method is devised, this fundamental scheme can accommodate extensions such as multi-agent collaborative learning and model learning.

Motivated by answers to the "why" question, we ask:

> *Can a single online agent learn the equilibria of mean field games efficiently?*

Our work affirmatively answers this question by presenting QM iteration (QMI), an efficient online single-agent model-free method for learning MFGs. QMI is strongly backed by the following theoretical premise. In an MFG, as all agents follow the same policy, we know that any agent's state is sampled from the population distribution. This fact reveals that a single agent encapsulates information about the entire population, suggesting that the induced population distribution can be learned through a single agent's state observations. More importantly, these observations are already collected during the phase where the agent updates its policy using an online RL method, suggesting that a single agent can learn both the BR and IP simultaneously using the same batch of online observations.

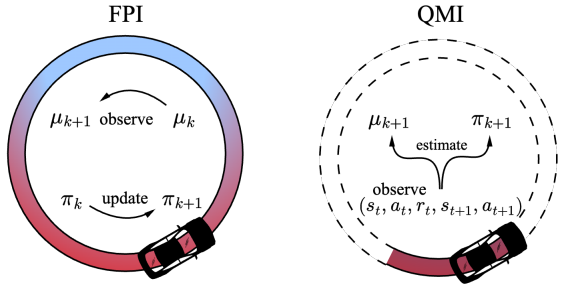


Figure 1: Illustration of learning processes of FPI and QMI for speed control on a ring road. The gradient color map signifies the varying population density on the ring road, with the dashed line indicating elements unobserved by the online agent. In FPI, the BR is calculated by a *representative* agent and the IP is directly observed. In QMI, a single online agent observes only *local* states and resultant rewards $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$, and uses these observations to estimate both the BR and IP.

We present the example of speed control on a ring road, as illustrated in Figure 1, to concretize the above ideas and highlight the improvements of QMI over FPI. In this game, vehicles aim to maintain some desired speed while avoiding collisions. In FPI, a *representative* agent interacts with the population mass to learn the BR. Then, a dedicated forward process is needed to calculate the IP, either by leveraging knowledge of the transition dynamics or directly observing population dynamics across the entire state space. In contrast, QMI employs a single online agent with only local state and reward observations. Unlike FPI's representative agent, the online agent in QMI has no population information and thus no interaction with the population mass. Consequently, it maintains an *estimate* of the IP, and derives rewards according to this estimate. Equipped with this estimate, the online agent in QMI, similar to FPI's representative agent, can update its policy using local observations by online RL methods. As a distinctive feature, this agent also uses these local observations to update its population distribution estimate. Hence, QMI consolidates the two separate backward and forward processes in FPI into one and eliminates the need for prior environmental knowledge and global communication.

Contributions. Our primary contributions include:

- We propose an online single-agent model-free scheme for learning MFGs, termed as QM iteration (QMI). At each step of QMI, the agent updates its BR and IP estimates *simultaneously* using an online observation. More practical than FPI, QMI is applicable when no prior knowledge of the transition dynamics or the state space is available. We develop two variants of QMI, contingent on whether the agent selects actions following a fixed *behavior* policy, or adaptively updates its behavior policy within an outer iteration (Algorithm 1). An overview of the distinct features exhibited by the two variants is provided in Table 1.
- We prove that QMI efficiently approximates FPI and, therefore enjoys a similar convergence guarantee. The resemblance between the learning dynamics of QMI and FPI is illustrated in Figure 2. We provide sample complexity guarantees for our methods (Theorem 1). Our methods are the first provably efficient online single-agent model-free methods for learning MFGs. We validate our findings through numerical experiments on various MFGs (Section 6 and Appendix B).

Related work. A comprehensive survey on the application of RL in learning MFGs is presented by Laurière et al. (2022); Cui et al. (2022). Existing work exclusively focuses on obtaining BRs in FPI using RL methods, including Q-learning (Guo et al., 2019; Perrin et al., 2021; Cui and Koeppl, 2021), policy gradient (Elie et al., 2020), and actor critic (Mguni et al., 2018; Subramanian et al., 2022; Chen et al., 2023b). To stabilize FPI, researchers have proposed various techniques,

including fictitious play (Cardaliaguet and Hadikhanloo, 2017; Perrin et al., 2020), online mirror descent (Perolat et al., 2021), and entropy regularization (Guo et al., 2022; Anahtarci et al., 2023). We include a more detailed discussion of related work in Appendix A.

2 Preliminaries

2.1 Mean Field Games

We consider an infinite-horizon discounted Markov decision process (MDP) denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite state and action spaces respectively, with their cardinality denoted by $S \coloneqq |\mathcal{S}|$ and $A \coloneqq |\mathcal{A}|$, r is the reward function, $\gamma \in (0,1)$ is the discount factor, and P is the transition kernel such that $P(s' \mid s, a)$ represents the probability that an agent transitions to state s' when it takes action a at state s. A policy (also referenced as a strategy or response) π maps a state to a distribution on the action space, guiding the action choices of an agent. When the policy π is fixed, we use P_π to denote the transition kernel and write $P_\pi(s, s') \coloneqq \sum_{a \in \mathcal{A}} P(s' \mid s, a)\pi(a \mid s)$.

In MFGs, agents are considered indistinguishable with individually negligible influence. Thus, an MFG encapsulates the impact of all agents on a given one through the concept of *population*. In this work, we consider reward functions that depend on the population distribution over the state space $\mu \in \Delta(\mathcal{S}) \coloneqq \{\text{distributions on } \mathcal{S}\}$. Specifically, a reward function $r : \mathcal{S} \times \mathcal{A} \times \Delta(\mathcal{S}) \to [0, R]$ signals a reward at each state-action pair based on the population distribution.

In MFGs, agents are rational and aim to maximize their expected cumulative reward. Our goal is to find an *optimal* policy—one that cannot be improved given that other agents' policies are fixed. We utilize a value-based approach to calculate policies. A Q-value function returns the expected cumulative reward starting from a state following the current policy π and population distribution μ:

$$Q_{\pi,\mu}(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, \mu\right) \,\middle|\, s_0 = s, a_0 = a\right], \quad (1)$$

where the expectation is taken w.r.t. the transition kernel P_π. Given a value function, we can choose the action accordingly, e.g., greedily select the action that maximizes the value function or use an ϵ-greedy selection. For broader adaptability, we presume access to a *policy operator* Γ_π that yields a policy based on a value function. Thus, the optimal policy can be characterized through a value function, translating our goal into discovering an optimal value function. We are now ready to present the optimality conditions.

Definition 1 (Mean field Nash equilibrium)**.** A value function-population distribution pair (Q, M) is a *mean field Nash equilibrium (MFNE)* if it satisfies:

$$\begin{cases} Q &= \mathcal{T}_{Q,M} Q, \\ M &= \mathcal{P}_Q M, \end{cases} \quad (2)$$

where $\mathcal{T}_{Q,M}$ is the Bellman operator:

$$\mathcal{T}_{Q,M} Q(s,a) = \mathbb{E}_Q[r(s,a,M) + \gamma Q(s', a')], \quad (3)$$

where $\mathbb{E}_Q$ denotes the expectation over $a, a' \sim \Gamma_\pi(Q)$ and $s' \sim P(\cdot \mid s, a)$; and $\mathcal{P}_Q$ is the transition operator:

$$\mathcal{P}_Q M(s') = \sum_{s \in \mathcal{S}} P_Q(s, s') M(s), \quad (4)$$

where we write $P_Q \coloneqq P_{\Gamma_\pi(Q)}$, as the policy is determined by the value function given a fixed policy operator.

In Definition 1, $Q \in \mathbb{R}^{S \times A}$ denotes a generic value function table, which is not necessarily an actual value function defined per (1). Similarly, $M \in \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is the probability simplex over $\mathcal{S}$, represents a generic population distribution which is not necessarily an actual policy-induced population distribution. Analogous to the Q-value function, we refer to this generic population distribution as the *M-value function*. We use subscripts, e.g., Q_M and μ_Q, to indicate actual BRs and IPs w.r.t. specific population distributions and value functions.

2.2 Fixed-Point Iteration for MFG

Fixed-point iteration (FPI), a classic method for learning MFGs, comprises two steps: evaluating the *best response* (BR) and the *induced population distribution* (IP). Fixing a population distribution M, the game reduces to a standard RL problem, which has a unique optimal value function (Bertsekas and Tsitsiklis, 1996), i.e., the BR w.r.t. the population distribution M. If the transition kernel P_Q yields a steady state distribution, this distribution is referred to as the IP w.r.t. the value function Q. Decomposing (2) gives formal definitions of these two operations.

Definition 2 (FPI operators)**.** The BR operator,

$$\Gamma_{\text{BR}} : \Delta(\mathcal{S}) \to \mathbb{R}^{S \times A}, \; M \mapsto Q_M,$$

returns the unique solution to the Bellman equation $Q_M = \mathcal{T}_{Q_M, M} Q_M$ for any population distribution M. The IP operator,

$$\Gamma_{\text{IP}} : \mathbb{R}^{S \times A} \to \Delta(\mathcal{S}), \; Q \mapsto \mu_Q,$$

returns the unique fixed point of the transition operator $\mathcal{P}_Q$ defined in (4) for any value function Q. Then, the FPI operator is the composition of the above two operators: $\Gamma \coloneqq \Gamma_{\text{IP}} \circ \Gamma_{\text{BR}} : \Delta(\mathcal{S}) \to \Delta(\mathcal{S})$.

Notably, the optimality in the BR is determined by the policy operator Γ_π. For example, When Γ_π is the greedy selector: $\Gamma_\pi^{(\max)}(Q)[a|s]=\mathbb{1}(a=\operatorname{argmax}_a Q(s,a))$, (3) becomes the Bellman optimality operator:

$$\mathcal{T}_M Q(s,a) = \mathbb{E}[r(s,a,M) + \gamma \max_{a'} Q(s',a')],$$

making BRs deterministic optimal policies. When Γ_π is the softmax function: $\Gamma_\pi^{(\text{softmax})}(Q)[a \mid s] = e^{LQ(s,a)} / \sum_{a'} e^{LQ(s,a')}$, where L is the inverse temperature parameter, the optimality corresponds to the MFG with entropy regularization (Cui and Koeppl, 2021; Guo et al., 2022; Anahtarci et al., 2023).

To focus on the main ideas, we consider *contractive* MFGs in this paper, where FPI is guaranteed to converge to the unique MFNE. Then, in Section 5, we show that our methods approximate FPI, thus enjoying a similar convergence guarantee. Without the contraction condition, stabilization techniques like fictitious play and online mirror descent need to be applied to FPI. We envision that our algorithms can be extended to incorporate these techniques with our analysis applying with minimal adjustment.

Assumption 1 (Contractive MFG)**.** The FPI operator is $(1-\kappa)$-contractive ($\kappa \in (0,1]$), i.e., for any $M_1, M_2 \in \Delta(\mathcal{S})$, it holds that

$$\|\Gamma M_1 - \Gamma M_2\|_2 \le (1-\kappa)\|M_1 - M_2\|_2.^1$$

3 Online Stochastic Updates

Without prior knowledge of the environment or the population, the online agent maintains two estimates—the Q-value function for the BR and the M-value function for the IP—which it updates using online stochastic observations. We first extend temporal difference (TD) control methods, a classic model-free RL framework covering Q-learning and SARSA (Sutton and Barto, 2018), to learn BRs, and then derive an online stochastic update rule for the IPs in the same vein.

Q-value function update. Guided by the Bellman operator (3), TD control gives an online stochastic update for the Q-value function:

$$\begin{aligned} Q(s,a) \leftarrow & Q(s,a) - \alpha g_Q(s,a,s',a'), \\ \text{with} \quad g_Q = & Q(s,a) - r(s,a,M) - \gamma Q(s',a'), \end{aligned} \tag{5}$$

where α is the step size, $s' \sim P(\cdot \mid s,a)$, and $a' \sim \Gamma_\pi(Q)$. If the policy operator is greedy and the behavior policy is fixed, the above update rule gives rise to off-policy Q-learning (Watkins and Dayan, 1992); for general policy operators, if the behavior policy updates in accordance to the value function, i.e., $\Gamma_\pi(Q)$ is the behavior policy and a' is the actual action, the above update rule gives rise to on-policy SARSA (Rummery and Niranjan, 1994; Singh and Sutton, 1996). We defer further discussion on these two TD control methods to Sections 4 and 5.

Population estimate. TD control replaces the expectation in the Bellman operator (3) with a stochastic approximation using online observations. Likewise, for the M-value function update, we first rewrite the transition operator using expectation:

$$\begin{aligned} \mathcal{P}_Q M(z) &= \sum_{s' \in \mathcal{S}} \delta_{s'}(z) \mathcal{P}_Q M(s') \\ &= \sum_{s' \in \mathcal{S}} \sum_{s \in \mathcal{S}} \delta_{s'}(z) P_Q(s,s') M(s) \\ &= \mathbb{E}_{Q,M}[\delta_{s'}(z)], \end{aligned} \tag{6}$$

where $\delta_{s'}$ is the indicator probability vector in $\Delta(\mathcal{S})$ such that $\delta_{s'}(z) = \mathbb{1}(z = s')$, and the expectation is taken over $s \sim M$ and $s' \sim P_Q(s,\cdot)$. Mimicking TD control and stochastic gradient descent, we remove the expectation and use the observed next successive state s' to stochastically approximate $\mathcal{P}_Q M$. This gives an online stochastic update for the M-value function:

$$M \leftarrow M - \beta g_M(s') = M + \beta(\delta_{s'} - M). \tag{7}$$

where β is the step size. The full derivation of this update rule is deferred to Appendix D. Similar to TD control, we anticipate that this update rule drives the M-value function to converge to the population distribution induced by P_Q. Furthermore, selecting a step size of $\beta_t = 1/(t+1)$ simplifies it to a Markov chain Monte Carlo (MCMC) method, validating its correctness.

For online stochastic updates (5) and (7) and general online learning methods to yield optimal solutions, the environment must be readily *explorable*. Unlike offline methods which rely on pre-collected data, an online agent learns and acts based on its real-time observations. Hence, the efficient learning of optimal policy becomes unfeasible if certain states are inaccessible, leading to potential suboptimal solutions. To avoid this, we impose the following condition on the MDP.

Assumption 2 (Ergodic MDP)**.** For any $Q \in \mathbb{R}^{S \times A}$, the Markov chain induced by the transition kernel P_Q is ergodic with a uniform mixing rate. In other words, there exists a steady state distribution μ_Q for any policy $\Gamma_\pi(Q)$, with constants $m \ge 1$ and $\rho \in (0,1)$, such that

$$\sup_{s \in \mathcal{S}} \sup_{Q \in \mathbb{R}^{S \times A}} \|P_\pi(S_t = \cdot \mid S_0 = s) - \mu_Q\|_{\text{TV}} \le m\rho^t.$$

[1] We consider L_1 and L_2 distances for probability measures in this work. For a finite state space with the trivial metric, the total variation distance equals the 1-Wasserstein distance (Gibbs and Su, 2002), with the L_1 distance being twice as large as them. Without loss of generality, we redefine the total variation distance as twice its standard definition, and use it interchangeably with the L_1 distance.

For future reference, we define an auxiliary constant $\sigma = \hat{n} + m\rho^{\hat{n}}/(1-\rho)$, where $\hat{n} = \lceil \log_\rho m^{-1} \rceil$. And the probability of visiting a state-action pair under a steady distribution is lower bounded:

$$\inf_{(s,a)\in\mathcal{S}\times\mathcal{A}} \inf_{Q\in\mathbb{R}^{S\times A}} \mu_Q(s)\cdot\Gamma_\pi(Q)[a\,|\,s] \geq \lambda_{\min} > 0.$$

Assumption 2 is a standard assumption for online RL methods (Bhandari et al., 2018; Zou et al., 2019; Qu and Wierman, 2020).

4 Proposed Methods

4.1 Off-Policy QM Iteration

Given online stochastic update rules (5) and (7), one can already construct a simple online single-agent model-free method by *iteratively* evaluating the BR and IP in the FPI method. However, a significant advantage of our online stochastic formulation of the update rules is that it enables *simultaneous* updates of both the Q-value and M-value functions using the same batch of observations.

Taking one step in this direction, we first present an algorithm that simultaneously evaluates both the BR and IP, with the agent's behavior policy being fixed within each outer iteration. Since the behavior policy is not updated along with the Q-value function, we use off-policy Q-learning to learn the BR, and term this method *off-policy* QM iteration. The method is presented in Algorithm 1 with input option `off-policy` and the greedy policy operator $\Gamma_\pi^{(\max)}$.

Our algorithm showcases marked simplicity. At each time-step, the online agent observes a state transition and a reward; it then uses this information to update the Q-value and M-value function tables using (5) and (7), respectively, which only involves two elementary operations—scaling and addition. It is noteworthy that in the Q-value function update, a_{t+1}, which follows the fixed behavior policy, is not used. Instead, $a' \sim \Gamma_\pi^{(\max)}(Q_{k,t})$ is used according to (5). The discrepancy accounts for the naming of "off-policy" Q-learning.

The stationary nature of the transition kernel within each outer iteration directly gives the convergence guarantee of off-policy QMI and suggests its analogy with FPI (see Section 5). Nevertheless, fixed transition kernels make off-policy QMI learn BRs and IPs *parallelly*. That is, at kth iteration, the BR w.r.t. $M_{k,0}$ is approximated by $Q_{k,T} = Q_{k+1,0}$, whose corresponding population distribution is then approximated by $M_{k+1,T} = M_{k+2,0}$, rather than $M_{k+1,0}$. Let $Q_k := Q_{k,0}$ and $M_k := M_{k,0}$. Then, off-policy QMI generates two non-interacting parallel policy-population sequences:

Algorithm 1: QM Iteration

input initial value functions $Q_{-1,T} = Q_0$ and $M_{-1,T} = M_0$; initial state s_0; option `off-policy` or `on-policy`
for $k = 0, 1, ..., K$ **do**
  $Q_{k,0} = Q_{k-1,T}, M_{k,0} = M_{k-1,T}$
  $\pi_{k,0} = \Gamma_\pi(Q_{k,0})$
  for $t = 0, 1, \ldots, T$ **do**
    sample one Markovian observation tuple $(s_t, a_t, s_{t+1}, a_{t+1})$ following policy $\pi_{k,t}$
    observe the reward $r(s_t, a_t, M_{k,0})$
    $Q_{k,t+1}(s_t, a_t) = Q_{k,t}(s_t, a_t) - \alpha_t g_{Q_{k,t}}$
    $M_{k,t+1} = M_{k,t} - \beta_t g_{M_{k,t}}(s_{t+1})$
    if `off-policy` **then**
      $\pi_{k,t+1} = \pi_{k,0}$
    else if `on-policy` **then**
      $\pi_{k,t+1} = \Gamma_\pi\left(\text{mix}\left(\{Q_{k,l}\}_{l=0}^{t+1}\right)\right)$
    end
  end
end
return $Q_{K,T}, M_{K,T}$

$\{(Q_{2k}, M_{2k+1})\}_{k=0}^{K/2}$ and $\{(Q_{2k+1}, M_{2k})\}_{k=0}^{K/2}$. This observation also implies that off-policy QMI is at least twice as efficient as FPI; see Figure 2 for an illustration.

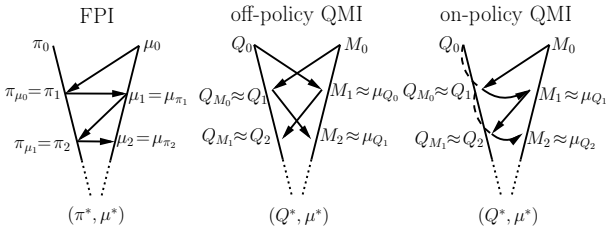


Figure 2: Illustration of learning processes. Each arrow represents one iteration in FPI or one outer iteration in QMI, matching the end BR or IP with the population distribution or value function at the start. The dashed line in on-policy QMI represents behavior policy updates, making M_k match the updated BR estimate Q_k.

To establish the convergence guarantee of off-policy QMI, we leverage the theoretical results of off-policy Q-learning. However, the greedy policy operator used is too *nonsmooth*: a slight difference in the Q-value function can lead to completely different action choices. As a result, the induced population distributions can drastically differ between outer iterations, leading to unstable convergence performance. Since we do not incorporate stabilization techniques, we make the following assumption.

Assumption 3 (Lipschitz continuous transition kernels

for Q-learning). For any $Q_1, Q_2 \in \mathbb{R}^{S\times A}$, it holds that

$$\|P_{Q_1} - P_{Q_2}\|_{\mathrm{TV}} \le L \|Q_1 - Q_2\|_2 ,$$

where $\|P_Q\|_{\mathrm{TV}} \coloneqq \sup_{\|q\|_{\mathrm{TV}}=1} \left\|\sum_{s\in\mathcal{S}} q(s) P_Q(s,\cdot)\right\|_{\mathrm{TV}}$.

4.2 On-Policy QM Iteration

Still, BR and IP evaluations are executed parallelly in off-policy QMI, and thus its efficiency boost indirectly attributes to parallel computing. This naturally raises a question: can we directly approximate the FPI operator Γ in one outer iteration? The *on-policy* variant of QM iteration provides a positive response. This time, we pass to Algorithm 1 the option `on-policy` and a general policy operator satisfying Assumption 4, facilitating dynamic updates of agent's behavior policy within each outer iteration. By syncing the behavior policy in accordance with the Q-value function, the policy learning process is governed by on-policy SARSA. Additionally, since the agent now observes the state transition induced by the updated policy, the M-value function is updated towards the population distribution induced by the updated policy. The learning process of on-policy QMI is illustrated in Figure 2.

On the other hand, constantly changing behavior policies in on-policy QMI yield nonstationary Markov chains. Such nonstationarity renders the convergence guarantee of off-policy QMI not applicable here and complicates the convergence analysis of on-policy QMI. Nonetheless, we establish a similar convergence guarantee for on-policy QMI (Lemma 3). To achieve the sharp logarithmic dependency on T, we *mix* the Q-value functions obtained in an outer iteration: $\mathrm{mix}\left(\{Q_{k,l}\}_{l=0}^{t+1}\right) \coloneqq \sum_{l=0}^{t}(w_l/\sum_{l=0}^{t} w_l) Q_{k,l}$, where $w_l \asymp t$, and use this convex combination to determine the behavior policy.

Theoretical results of on-policy SARSA are used to establish the convergence guarantee of on-policy QMI. While the instability issue persists as in off-policy QMI, on-policy SARSA's adaptability and versatility—facilitated by its use of general policy operators—outstrip those of Q-learning, thus allowing us to directly impose the smoothness condition on Γ_π.

Assumption 4 (Lipschitz continuous policy operator for SARSA)**.** For any $Q_1, Q_2 \in \mathbb{R}^{S\times A}$ and $s \in \mathcal{S}$, it holds that

$$\|\Gamma_\pi(Q_1)[\cdot \mid s] - \Gamma_\pi(Q_1)[\cdot \mid s]\|_{\mathrm{TV}} \le L\|Q_1 - Q_2\|_2.$$

Furthermore, the Lipschitz constant satisfies $L \le \lambda_{\min}(1-\gamma)^2/(2R\sigma)$.

Remark 1. Assumption 3 is weaker than Assumption 4 as the latter implies the former (see Lemma 4) and requires a small Lipschitz constant. However, verifying Assumption 3 can be difficult as the dependence of P_Q on Q can be intricate and the model is unknown, whereas Assumption 4 is more achievable given the flexibility in choosing policy operators for SARSA. For instance, the softmax function with an apt temperature parameter satisfies Assumption 4 (Gao and Pavel, 2017). Actually, a softmax policy operator imposes entropy regularization to the greedy selection (Gao and Pavel, 2017), a common technique used to stabilize the MFG learning process (Cui and Koeppl, 2021; Guo et al., 2022; Anahtarci et al., 2023). Absent such regularization, Assumption 3 ensures training stability. Other assumptions have been posited for this purpose, such as a strongly convex Bellman operator (Anahtarci et al., 2019). Notably, Assumption 1 and 3 or 4 are not mutually exclusive; either Assumption 3 or 4 with some conditions on the reward function's smoothness and Lipschitz constants can imply Assumption 1 (Guo et al., 2019; Anahtarci et al., 2019).

4.3 Comparison of Off- and On-Policy QMI

Table 1: Comparison of off- and on-policy QMI.

	Off-Policy	**On-Policy**
Behavior policy within an outer iteration	Fixed	Adaptive
Robustness	✗	✓
MFNE	original	regularized
Sample efficiency boost mechanism	parallel	concurrent
Population-dependent transition kernels	✗	✓

Table 1 gives an overview of the differences between off- and on-policy variants of QMI. By utilizing Q-learning with a greedy policy operator, off-policy QMI can learn a deterministic optimal policy of the original MFG. On-policy QMI, on the other hand, utilizes SARSA with a *soft* (non-deterministic) policy operator, meaning that the learned MFNE depends on the policy operator and corresponds to a regularized MFG. Nevertheless, on-policy QMI affords flexibility in choosing a wider range of policy operators, and the soft policies it acquires exhibit greater robustness (Sutton and Barto, 2018). Furthermore, off-policy QMI boosts the sample efficiency by learning two policy-population sequences parallelly, while on-policy QMI directly boosts it by amalgamating the two steps in FPI into one. Last but not least, on-policy QMI and its convergence guarantee can directly accommodate transition kernels that are dependent not only on behavior policy but also on population distribution. However, such a dependence breaks the parallel procedure in off-policy QMI. See Appendix G.1 for a detailed discussion on population-dependent transition kernels.

5 Sample Complexity Analysis

In this section, we establish the sample complexity guarantee for both the off- and on-policy variants of Algorithm 1, given our assumptions on MDPs (Assumption 2), MFGs (Assumption 1), and smoothness (Assumption 3 or Assumption 4). To assist the analysis, we define the operators presented in Algorithm 1, which correspond to those in Definition 2.

Definition 3 (QMI operators)**.** For off-policy QMI, the Q- and M-value function operators,

$$\Gamma_Q(T) : \Delta(\mathcal{S}) \to \mathbb{R}^{S\times A},\ M_{k,0} \mapsto Q_{k,T} \quad \text{and}$$
$$\Gamma_M(T) : \mathbb{R}^{S\times A} \to \Delta(\mathcal{S}),\ Q_{k,0} \mapsto M_{k,T},$$

return the updated Q- and M-value function after an outer iteration of Algorithm 1, consisting of T online stochastic updates using Lines 8 and 9. Then, the off-policy QMI operator is the composition of the above two operators: $\hat{\Gamma}_{\text{off}}(T) \coloneqq \Gamma_M(T) \circ \Gamma_Q(T)$.

The on-policy QMI operator,

$$\hat{\Gamma}_{\text{on}}(T) : \Delta(\mathcal{S}) \to \Delta(\mathcal{S}),\ M_{k,0} \mapsto M_{k,T},$$

returns the updated M-value function after an outer iteration of Algorithm 1, consisting of T online stochastic updates using Lines 8 and 9, as well as the policy updates using Line 13.

As mentioned in previous sections, the equivalence between off-policy QMI and FPI comes from the convergence of off-policy Q-learning and MCMC. Specifically, we have the following two lemmas.

Lemma 1 (Sample complexity of Q-learning (Qu and Wierman, 2020, Theorem 7))**.** *Suppose Assumptions 2 and 3 hold for the greedy policy operator. With a step size of $\alpha_t \asymp 1/(\lambda_{\min}(1-\gamma)t)$, for any $M \in \Delta(\mathcal{S})$, we have*

$$\mathbb{E}\left\|\Gamma_Q(T)M - \Gamma_{\text{BR}}M\right\|_2^2 = O\left(\frac{SAR^2\log T}{\lambda_{\min}^2(1-\gamma)^5 T}\right),$$

where MDP components $\mathcal{S}, \mathcal{A}, R$, and γ are defined in Section 2.1, with S and A denote the cardinality of $\mathcal{S}$ and $\mathcal{A}$. σ and $\lambda_{\min}$ are defined in Assumption 2, and L is defined in Assumption 3.

Lemma 2 (Sample complexity of stationary MCMC (Łatuszyński et al., 2013, Theorem 3.1))**.** *Suppose Assumption 2 holds. With a step size of $\beta_t \asymp 1/t$, for any $Q \in \mathbb{R}^{S\times A}$, we have*

$$\mathbb{E}\left\|\Gamma_M(T)Q - \Gamma_{\text{IP}}Q\right\|_2^2 = O\left(\frac{SA}{(1-\rho)^2 T}\right).$$

The preceding lemmas demonstrate that off-policy QMI efficiently approximates FPI, with the Q-value and M-value updates evaluating the BR and IP, respectively.

Lemmas 1 and 2 are not applicable to on-policy QMI, where transition kernels are nonstationary. Nonetheless, we can establish the following lemma.

Lemma 3 (Sample complexity of nonstationary MCMC with SARSA)**.** *Suppose Assumptions 2 and 4 hold for the chosen policy operator. With a step size of $\alpha_t \asymp 1/(\lambda_{\min}(1-\gamma)t)$ and $\beta_t \asymp 1/t$, for any $M \in \Delta(\mathcal{S})$, we have*

$$\mathbb{E}\left\|\hat{\Gamma}_{\text{on}}(T)M - \Gamma M\right\|_2^2 = O\left(\frac{SAR^2L^2\sigma^2\log T}{\lambda_{\min}^2(1-\gamma)^4 T}\right).$$

An outer iteration of on-policy QMI corresponds to a nonstationary MCMC. Thus, to prove Lemma 3, we employ a *backtracking* procedure, a technique developed in Zou et al. (2019) to address nonstationarity in stochastic approximation methods. The key idea is that in order to exploit the mixing property of stationary Markov chains (Assumption 2), we virtually backtrack a period τ, and generate a virtual trajectory where the agent follows the fixed behavior policy $\pi_{t-\tau} \coloneqq \Gamma_\pi(Q_{t-\tau})$ after time step $t-\tau$. This virtual trajectory is stationary after time step $t-\tau$ and rapidly converges to the steady distribution induced by $\pi_{t-\tau}$, denoted by $\mu_{t-\tau}$. Next, the convergence of SARSA confirms that $\mu_{t-\tau}$ converges to the steady distribution induced by the BR w.r.t. M, denoted by $\mu_M \coloneqq \Gamma M$. Let s_t and $\tilde{s}_t$ be the state at time step t on the actual and virtual trajectories, respectively. Let π_t be the (actual) behavior policy at time step t, with its induced steady distribution denoted as μ_t. Then, the proof sketch for Lemma 3 can be succinctly portrayed as:

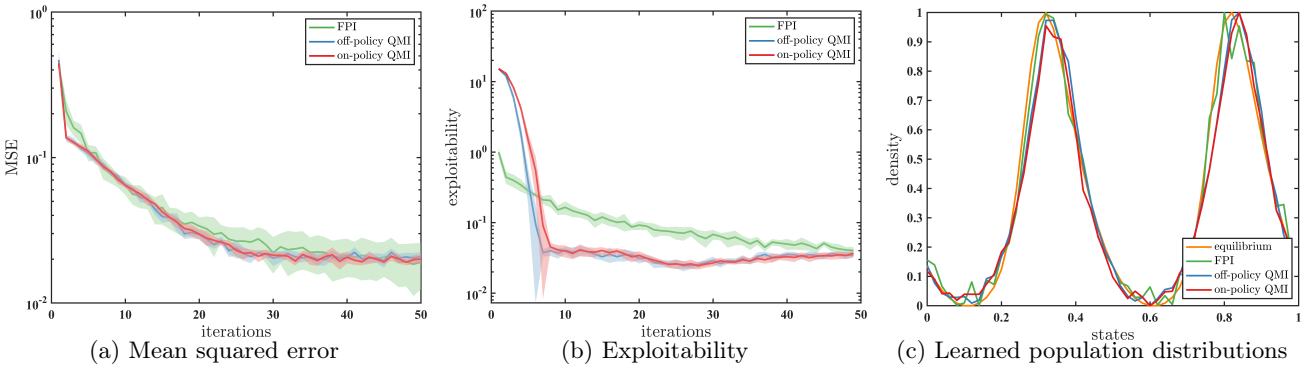
$$\underbrace{s_t \approx \overbrace{\tilde{s}_t \xrightarrow[\tau\to\infty]{d} s}^{H_2,\text{mix}}}_{H_1,\text{backtrack}} \sim \underbrace{\mu_{t-\tau} \approx \overbrace{\mu_t \xrightarrow[t\to\infty]{L_2} \mu_M}^{H_4,\text{SARSA}}}_{H_3,\text{progress}},$$

where the backtracking discrepancy H_1 and the distribution progress H_3 are controlled by the virtual period τ (Lemmas 8 and 10), while the two convergence rates H_2 and H_4 are characterized by the geometric ergodicity of stationary MDPs and the sample complexity of SARSA (Lemmas 7 and 9), respectively. In brief, we show that the agent's state distribution, and thus its M-value function, rapidly converges to the IP μ_M.

Given the above lemmas, we are ready to compose the convergence guarantee and sample complexity of QMI.

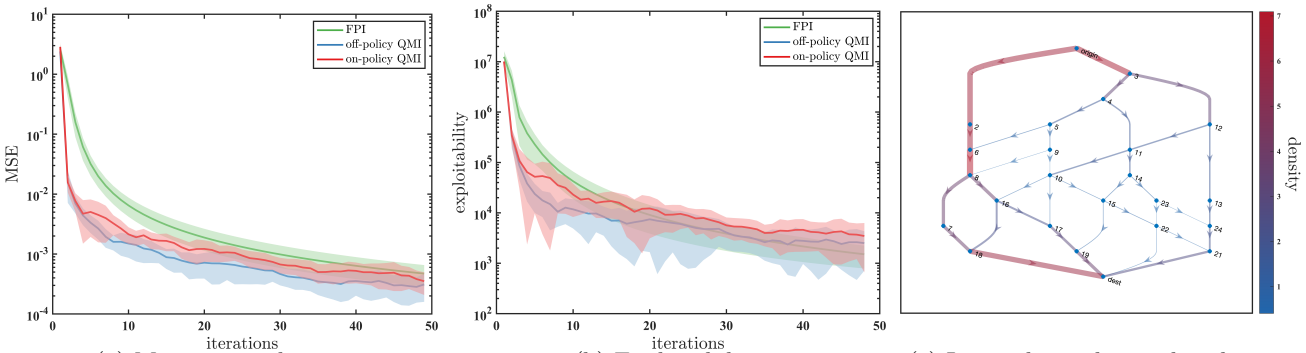
Theorem 1 (Sample complexity of QMI)**.** *Suppose Assumptions 1 and 2 hold, and Assumptions 3 and 4 hold for off- and on-policy QMI, respectively. Let μ^* be the MFNE population distribution. Then Algorithm 1 returns an ϵ-approximate MFNE, that is,*

$$\mathbb{E}\left\|M_{K,T} - \mu^*\right\|_2^2 = \mathbb{E}\|\hat{\Gamma}(T)^K M_0 - \mu^*\|_2^2 \le \epsilon^2,$$



(a) Mean squared error (b) Exploitability (c) Learned population distributions

Figure 3: Performance comparison of FPI, off-policy QMI, and on-policy QMI on ring road speed control. MSE represents the mean squared L_2 error between the current M-value function and the MFNE population distribution. Exploitability refers to the disparity between the current value function and the BR w.r.t. the current population distribution (see Perrin et al. (2020)). Learned population distributions are scaled for better visualization.



(a) Mean squared error (b) Exploitability (c) Learned population distribution

Figure 4: Performance comparison of FPI, off-policy QMI, and on-policy QMI on Sioux Falls network routing. Only the population distribution learned by off-policy QMI is shown in (c); other methods and the MFNE give similar population distributions and are deferred to Appendix B.1.

with the number of iterations being at most

$$K = O\left(\kappa^{-1}\log\epsilon^{-1}\right), \quad T = C\cdot O\left(\kappa^{-2}\epsilon^{-2}\log\epsilon^{-1}\right),$$

where

$$C \le \frac{SAR^2L^2\sigma^2}{\lambda_{\min}^2(1-\gamma)^5}.$$

Our complexity results match the prior work on learning MFGs (Anahtarci et al., 2023) and are consistent with stochastic approximation methods (Łatuszyński et al., 2013; Zou et al., 2019; Qu and Wierman, 2020).

6 Numerical Experiments

Speed control on a ring road. We consider a speed control game of autonomous vehicles on a ring road, the example presented in Figure 1. We design the following cost function for this goal based on the Lighthill-Whitham-Richards function:

$$r(s,a,\mu) = -\frac{1}{2}\left(b(s) + \frac{1}{2}\left(1 - \frac{\mu(s)}{\mu_{\text{jam}}}\right) - \frac{a}{a_{\max}}\right)^2 \Delta s,$$

where b encodes the location preference, μ_{jam} is the jam density, and $a_{\max}$ is the maximum speed. The performance comparison is reported in Figure 3.

Routing game on a network. We consider a routing game on the Sioux Falls network, a graph with 24 nodes and 74 directed edges. On each edge, a vehicle needs to select its next edge to travel to. Therefore, both the state space and the action space are the edge set, i.e., $\mathcal{S} = \mathcal{A} = \{e_1, \ldots, e_{74}\}$. The objective of a vehicle is to reach the destination as fast as possible. Due to congestion, a vehicle spends a longer time on an edge with higher population distribution, giving the following reward function:

$$r(s,a,\mu) = \underbrace{-c_1\mu(s)^2\mathbb{1}\{s \neq e_{74}\}}_{\text{congestion cost}} + \underbrace{c_2\mathbb{1}\{s = e_{74}\}}_{\text{terminal reward}}.$$

The performance comparison is reported in Figure 4.

All numerical results are averaged over 10 independent runs. Please refer to Appendix B for the full setups of two experiments and additional numerical results.

References

B. Anahtarci, C. D. Kariksiz, and N. Saldi. Fitted Q-learning in mean-field games. *arXiv preprint arXiv:1912.13309*, 2019.

B. Anahtarci, C. D. Kariksiz, and N. Saldi. Q-learning in regularized mean-field games. *Dynamic Games and Applications*, 13(1):89–117, 2023.

Anonymous. Finite-time analysis of on-policy heterogeneous federated reinforcement learning. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=D2e0VqPX9g. under review.

D. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

J. Bhandari, D. Russo, and R. Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692. PMLR, 2018.

P. Cardaliaguet and S. Hadikhanloo. Learning in mean field games: The fictitious play. *ESAIM: Control, Optimisation and Calculus of Variations*, 23(2):569–591, 2017.

R. Carmona, M. Laurière, and Z. Tan. Model-free mean-field reinforcement learning: mean-field mdp and mean-field Q-learning. *arXiv preprint arXiv:1910.12802*, 2019.

X. Chen, S. Liu, and X. Di. Learning dual mean field games on graphs. In *Proceedings of the 2023 European Conference on Artificial Intelligence*, 09 2023a.

X. Chen, S. Liu, and X. Di. A hybrid framework of reinforcement learning and physics-informed deep learning for spatiotemporal mean field games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1079–1087, 2023b.

K. Cui and H. 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.

K. Cui, A. Tahir, G. Ekinci, A. Elshamanhory, Y. Eich, M. Li, and H. Koeppl. A survey on large-population systems and scalable multi-agent reinforcement learning. *arXiv preprint arXiv:2209.03859*, 2022.

R. Elie, J. Perolat, M. Laurière, M. Geist, and O. Pietquin. On the convergence of model free learning in mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7143–7150, New York, NY, USA, 04 2020. AAAI.

B. Gao and L. Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

X. Guo, A. Hu, R. Xu, and J. Zhang. Learning mean-field games. *Advances in Neural Information Processing Systems*, 32, 2019.

X. Guo, R. Xu, and T. Zariphopoulou. Entropy regularization for mean field games with learning. *Mathematics of Operations research*, 47(4):3239–3260, 2022.

M. Huang, R. P. Malhamé, and P. E. Caines. Large population stochastic dynamic games: Closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information and Systems*, 6(3):221–252, 2006.

J.-M. Lasry and P.-L. Lions. Mean field games. *Japanese journal of mathematics*, 2(1):229–260, 2007.

K. Łatuszyński, B. Miasojedow, and W. Niemiro. Nonasymptotic bounds on the estimation error of MCMC algorithms. *Bernoulli*, 19(5A):2033–2066, 2013.

M. Laurière, S. Perrin, M. Geist, and O. Pietquin. Learning mean field games: A survey. *arXiv preprint arXiv:2205.12944*, 2022.

M. Lauriere, S. Perrin, S. Girgin, P. Muller, A. Jain, T. Cabannes, G. Piliouras, J. Perolat, R. Elie, O. Pietquin, and M. Geist. Scalable Deep Reinforcement Learning Algorithms for Mean Field Games. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12078–12095. PMLR, June 2022.

K. Lee, D. Rengarajan, D. Kalathil, and S. Shakkottai. Reinforcement learning for mean field games with strategic complementarities. In *International Conference on Artificial Intelligence and Statistics*, pages 2458–2466. PMLR, 2021.

G. Li, C. Cai, Y. Chen, Y. Wei, and Y. Chi. Is Q-learning minimax optimal? a tight sample complexity analysis. *Operations Research*, 2023.

S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer Science & Business Media, 2012.

D. Mguni, J. Jennings, and E. M. de Cote. Decentralised learning in systems with many, many strategic agents. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

A. Yu. Mitrophanov. Sensitivity and convergence of uniformly ergodic markov chains. *Journal of Applied Probability*, 42(4):1003–1014, 2005. ISSN 0021-9002.

J. Perolat, S. Perrin, R. Elie, M. Laurière, G. Piliouras, M. Geist, K. Tuyls, and O. Pietquin. Scaling up mean field games with online mirror descent. *arXiv preprint arXiv:2103.00623*, 2021.

S. Perrin, J. Pérolat, M. Laurière, M. Geist, R. Elie, and O. Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. *Advances in Neural Information Processing Systems*, 33:13199–13213, 2020.

S. Perrin, M. Laurière, J. Pérolat, M. Geist, R. Élie, and O. Pietquin. Mean field games flock! The reinforcement learning way. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 356–362. International Joint Conferences on Artificial Intelligence Organization, 2021.

G. Qu and A. Wierman. Finite-time analysis of asynchronous stochastic approximation and Q -learning. In *Conference on Learning Theory*, pages 3185–3205. PMLR, 2020.

G. A. Rummery and M. Niranjan. *On-Line Q-learning Using Connectionist Systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

Z. Shou, X. Chen, Y. Fu, and X. Di. Multi-agent reinforcement learning for markov routing games: A new modeling paradigm for dynamic traffic assignment. *Transportation Research Part C: Emerging Technologies*, 137:103560, 2022.

S. P. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine learning*, 22 (1-3):123–158, 1996.

S. Subramanian, M. Taylor, M. Crowley, and P. Poupart. Decentralized mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

J. Yang, X. Ye, R. Trivedi, H. Xu, and H. Zha. Learning deep mean field games for modeling large population behavior. *arXiv preprint arXiv:1711.03156*, 2017.

S. Zou, T. Xu, and Y. Liang. Finite-sample analysis for sarsa with linear function approximation. *Advances in neural information processing systems*, 32, 2019.

Appendix

A Related Work

Fixed-point iteration. Huang et al. (2006) introduced mean field games and suggested a forward-backward fixed-point iteration method to solve them. Guo et al. (2019) proposed using Q-learning for the backward best response calculation in fixed-point iteration. The theoretical assumptions needed for FPI are often difficult to verify, and empirically, FPI tends to exhibit instability in discrete-time (Cui and Koeppl, 2021). To stabilize FPI, researchers have proposed various techniques, including fictitious play (Cardaliaguet and Hadikhanloo, 2017; Perrin et al., 2020), online mirror descent (Perolat et al., 2021), and entropy regularization (Cui and Koeppl, 2021; Guo et al., 2022; Anahtarci et al., 2023). Lauriere et al. (2022) extended these techniques to incorporate scalable deep neural networks. In addition to FPI, other formulations have been proposed to avoid solving a forward-backward process. Yang et al. (2017); Chen et al. (2023a) formulated MFGs as a population MDP, which requires the knowledge of the entire state space and transition dynamics to update the state of the population.

Learning MFGs. A comprehensive survey on the application of RL in learning MFGs is presented by Laurière et al. (2022); Cui et al. (2022). Existing work exclusively focuses on obtaining BRs in FPI using RL methods. Q-learning is the most widely used method for learning the BRs (Guo et al., 2019; Perrin et al., 2021; Cui and Koeppl, 2021; Anahtarci et al., 2019, 2023). Other methods such as policy gradient (Elie et al., 2020; Carmona et al., 2019) and actor critic (Mguni et al., 2018; Subramanian et al., 2022; Chen et al., 2023b) have also been employed to learn the BRs. All existing methods require either knowledge of the transition dynamics or the direct observability of the population evolution.

B Additional Experiments

B.1 Speed Control on a Ring Road

We consider a speed control game of autonomous vehicles on a ring road, i.e., the unit circle $\mathbb{S}^1 \cong [0, 1)$, as illustrated in Figure 1. At location $s \in \mathbb{S}^1$, the representative vehicle selects a speed a, and then moves to the next location following transition $s' = s + a\Delta t \pmod 1$, where Δt is the time interval between two consecutive decisions. Without loss of generality, we assume that the speed is bounded by 1, i.e., the speed space is also $[0, 1)$. Then we discretize both the location space and the speed space using a granularity of $\Delta s = \Delta a = 0.02$. Thus, both our discretized state (location) space and action (speed) space can be represented by $\mathcal{S} = \mathcal{A} = \{0, 0.02, \ldots, 0.98\} \cong [50]$. By the Courant-Friedrichs-Lewy condition, we choose the time interval to be $\Delta t = 0.02 \leq \Delta s / \max a$. The objective of a vehicle is to maintain some desired speed while avoiding collisions with other vehicles. Thus, it needs to reduce the speed in areas with high population density. A classic cost function for this goal is the Lighthill-Whitham-Richards function:

$$r^{(\mathrm{LWR})}(s, a, \mu) = -\frac{1}{2}\left(\left(1 - \frac{\mu(s)}{\mu_{\mathrm{jam}}}\right) - \frac{a}{a_{\max}}\right)^2 \Delta s,$$

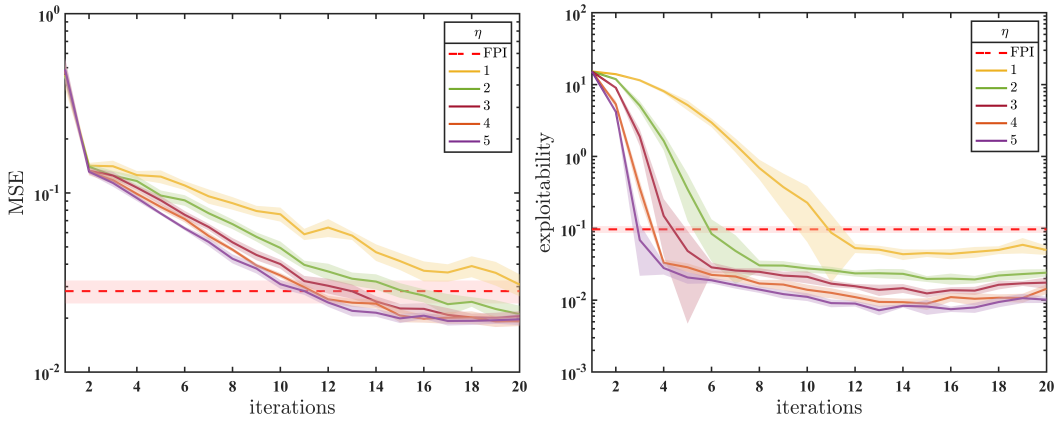
where μ_{jam} is the jam density, and $a_{\max}$ is the maximum speed. However, in an infinite horizon game, this cost function induces a *trivial* MFNE, where the equilibrium policy and population are both constant across the state space. Therefore, we introduce a stimulus term b that varies across different locations:

$$r(s, a, \mu) = -\frac{1}{2}\left(b(s) + \frac{1}{2}\left(1 - \frac{\mu(s)}{\mu_{\mathrm{jam}}}\right) - \frac{a}{a_{\max}}\right)^2 \Delta s,$$

where the factor of one-half before the population distribution term is included to account for the presence of the new stimulus term. This new cost function makes the MFNE more complex and corresponds to real-world situations where vehicles may have distinct desired speeds at different locations due to environmental variations. Specifically, we choose the stimulus term as $b(s) = 0.2(\sin(4\pi s) + 1)$, and set $\mu_{\mathrm{jam}} = 3/S$ and $a_{\max} = 1$.

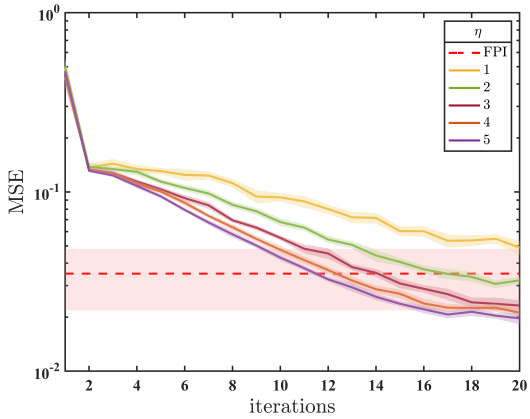
We compare QMI with model-based FPI: the BRs are calculated using value iteration (Sutton and Barto, 2018):

$$V_{t+1}(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a, \mu) + \gamma \sum_{s'} P(s' \mid s, a) V_t(s') \right\}, \tag{8}$$

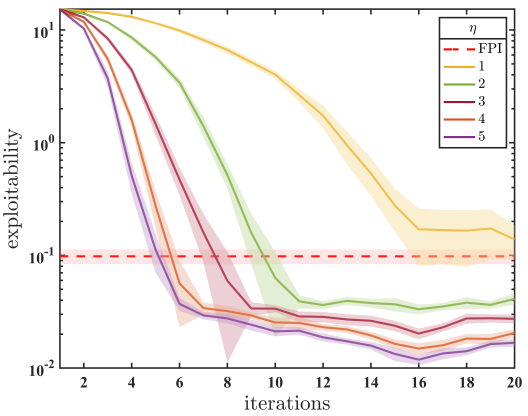


(a) Mean squared error of off-policy QMI

(b) Exploitability of off-policy QMI



(c) Mean squared error of on-policy QMI



(d) Exploitability of on-policy QMI

Figure 5: Performance comparison of different sample compensation factors on ring road speed control. As a baseline, the performance of FPI after 20 iterations is plotted as dashed lines.

and the induced population distributions are directly calculated using (4). Model-based FPI assumes full knowledge of the state-action space, reward function, as well as transition dynamics. During each iteration of value iteration and population update using (4), all S values are updated without any random sampling, and we refer to such an iteration as a *sweep*. It is expected that in order for online sampling to replicate the effects of a sweep, the number of samples should be at least S. Furthermore, since QMI assumes no knowledge of the action space and the reward function, it may require A samples to achieve the same effect as the max calculation in (8). The randomness in sampling can further impact efficiency. Therefore, we introduce a *sample compensation factor* η to relate the number of samples to the number of sweeps. Specifically, let T_{QMI} and T_{FPI} be the number of inner iterations of QMI (Algorithm 1) and the number of sweeps of value iteration and population update in FPI respectively; we let

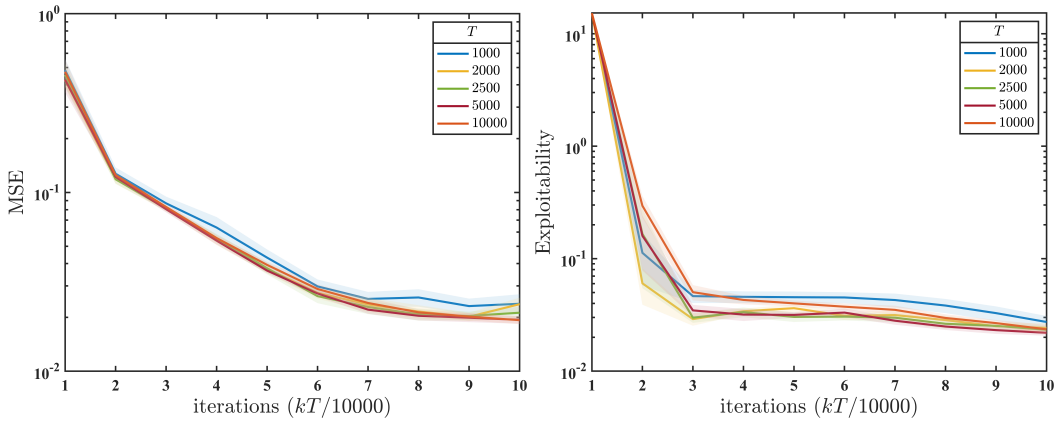
$$T_{\text{QMI}} = \eta S T_{\text{FPI}}.$$

We focus on two primary metrics: the mean squared error (MSE) of the population distribution and the exploitability of the policy. For a finite state space with the trivial metric, the total variation distance equals the 1-Wasserstein distance (Gibbs and Su, 2002), and is equivalent to the Euclidean norms. Thus, we consider the L_2 MSE between the current M-value function and the MFNE population distribution:

$$\text{MSE}(M) \coloneqq \|M_k - \mu^*\|_2^2 = \sum_{s \in \mathcal{S}} \left(M_k(s) - \mu^*(s)\right)^2.$$

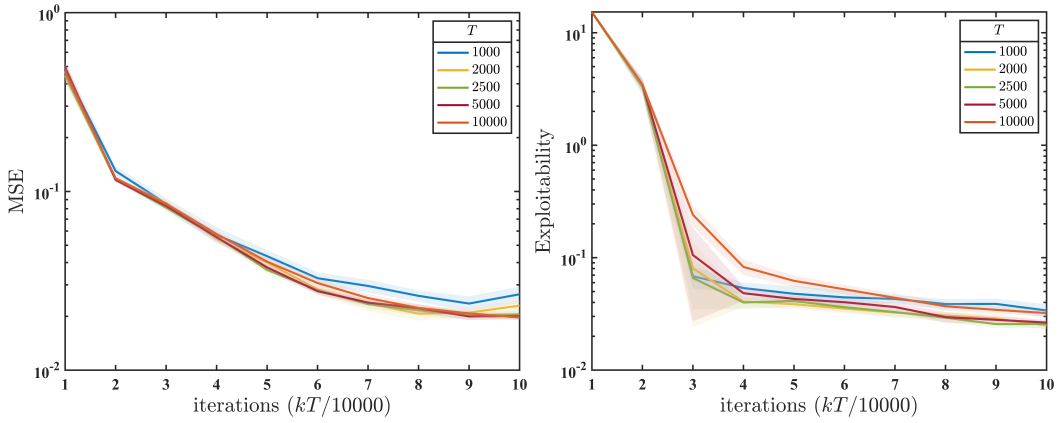
Perrin et al. (2020) defines the exploitability of a policy as follows:

$$\text{exploitability}(\pi) \coloneqq \max_{\pi'} \mathbb{E}_{s \sim \mu_\pi} V(s; \pi', \mu_\pi) - \mathbb{E}_{s \sim \mu_\pi} V(s; \pi, \mu_\pi),$$



(a) Mean squared error of off-policy QMI

(b) Exploitability of off-policy QMI



(c) Mean squared error of on-policy QMI

(d) Exploitability of on-policy QMI

Figure 6: Performance comparison of different number of inner iterations given a fixed total sample size: $KT = 10^5$ on ring road speed control.

where $V(s; \pi, \mu)$ is the value function determined by policy π and population distribution μ. Given a policy operator, the exploitability quantifies the gap between the current value function Q and the BR w.r.t. the population distribution induced by Q. We denote this BR by Q_{μ_Q}, and calculate $\|Q_{\mu_Q} - Q\|$ as the exploitability in practice.

The other algorithmic parameters are chosen as follows: the discount factor is set as $\gamma = 1 - \Delta s = 0.98$, the initial value function is set as all-zero, the initial state and initial population distribution is randomly generated. We use the softmax function with an increasing temperature of $50k$ as the policy operator in the kth outer iteration. The effective number of outer iterations is $K = 50$ and the number of sweeps in FPI is $T_{\text{FPI}} = 20$. All the results are averaged over 10 independent runs.

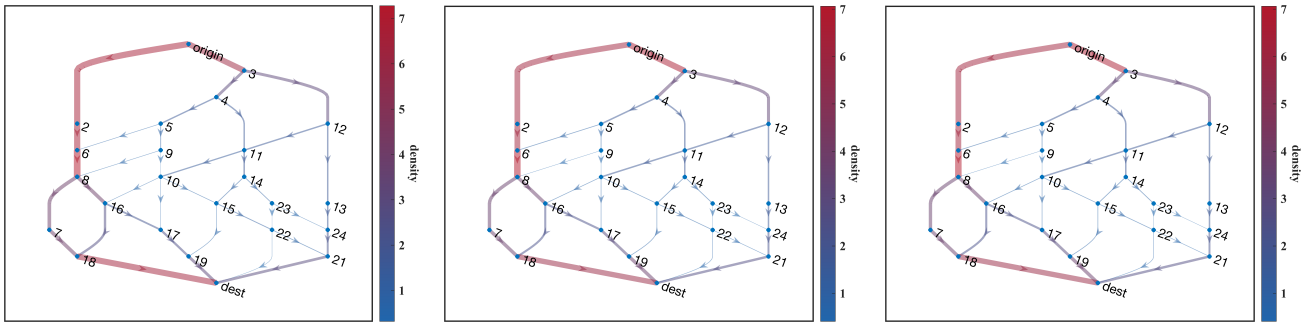
With a sample compensation factor of $\eta_{\text{off}} = 2$ for off-policy QMI and $\eta_{\text{on}} = 3$ for on-policy QMI, both variants achieve a similar efficacy as QMI, validating our methods. The results are reported in Figure 3.

We also experiment with different sample compensation factors. The results are reported in Figure 5. As we can see, both MSE and exploitability decrease as the sample compensation factor increases, which is expected since a larger T_{QMI} leads to a more accurate approximation of FPI. Furthermore, the improvement plateaus for large sample compensation factors, suggesting that a small sample compensation factor is sufficient, which is also more sample-efficient.

The numerical results of the experiments on the sample compensation factor confirm the intuition: as T_{QMI} increases, performance improves since the QMI operators more accurately approximate the FPI operators. However, a larger T_{QMI} necessitates more samples, and thus the sample efficiency decreases. Therefore, we also investigated the impact of T_{QMI} given a fixed total number of samples $K \cdot T$. The results are reported in

Figure 6. In this experiment, we use a constant step size of $\alpha_t = 0.001$ for Q-value function updates. The results demonstrate that different numbers of inner iterations offers nearly identical performance, implying that QMI is robust against the inexactness of the BR and IP approximation, as long as a sufficient total number of samples is present.

B.2 Routing Game on a Network



(a) On-policy QMI (b) FPI (c) MFNE

Figure 7: Learned population distributions and MFNE population distribution.

We consider a routing game on the Sioux Falls network,[2] a graph with 24 nodes and 74 directed edges. We designate node 1 as the starting point and node 20 as the destination. To construct an infinite-horizon game, we add a *restart* edge e_{75} from the destination back to the starting point. On each edge, a vehicle selects its next edge to travel to. We consider a deterministic environment, meaning that the vehicle will follow the chosen edge without any randomness. Therefore, both the state space and the action space can be represented by the edge set, i.e., $\mathcal{S} = \mathcal{A} = \{e_1, \dots, e_{75}\} \cong [75]$, where e_{75} is the restart edge. It is worth noting that a vehicle can only select from the outgoing edges of its current location as its next edge.

The objective of a vehicle is to reach the destination as fast as possible. Due to congestion, a vehicle spends a longer time on an edge with higher population distribution. Specifically, the cost (time) on a non-restart edge is $r^{(\text{cong.})}(s, a, \mu) = -c_1 \mu(s)^2 \mathbb{1}\{s \neq e_{75}\}$, where c_1 is a cost constant. To drive the vehicle to the destination, we impose a reward at the restart edge: $r^{(\text{term.})}(s, a, \mu) = c_2 \mathbb{1}\{s = e_{75}\}$. Together, we get the cost function:

$$r(s, a, \mu) = \underbrace{-c_1 \mu(s)^2 \mathbb{1}\{s \neq e_{75}\}}_{\text{congestion cost}} + \underbrace{c_2 \mathbb{1}\{s = e_{75}\}}_{\text{terminal reward}}.$$

We set $c_1 = 10^5$ and $c_2 = 10$. The other algorithmic parameters are chosen as follows: the discount factor $\gamma = 0.8$, the initial state is uniformly sampled, the initial value function is set as all-zero, the initial population is randomly generated. We use the softmax function with a fixed temperature of 10^{-4} as the policy operator. The effective number of outer iterations is $K = 50$ and the number of sweeps in FPI is $T_{\text{FPI}} = 5$. All the results are averaged over 10 independent runs.

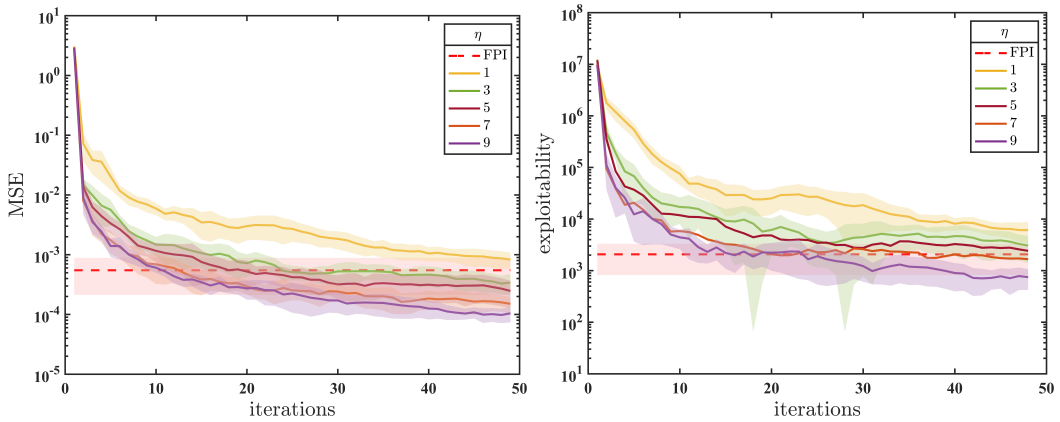
With a sample compensation factor of $\eta_{\text{off}} = 4$ for off-policy QMI and $\eta_{\text{on}} = 7$ for on-policy QMI, both variants achieve a similar efficacy as QMI, again, validating our methods. The results are reported in Figure 4. The population distributions learned by on-policy QMI and FPI, as well as the MFNE population distribution, are shown in Figure 7.

Similar to Appendix B.1, we also experiment with different sample compensation factors and different number of inner iterations given a fixed total number of samples. The results are reported in Figures 8 and 9. The results are consistent with those in Appendix B.1, reinforcing the validation, efficiency, and robustness of our methods.

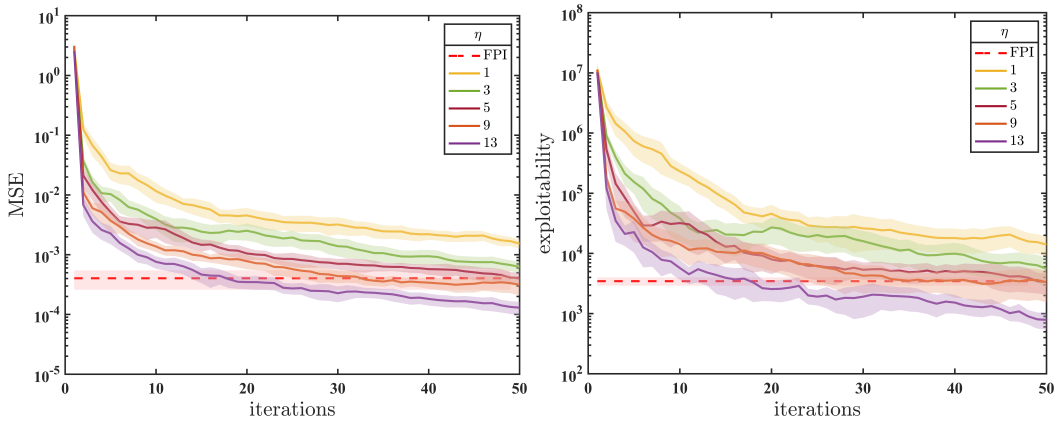
C Notation

We summarize the notations used in this paper in Table 2. Additionally, the L_2 norm is used when we omit the subscript. For value function sequences generated by Algorithm 1, we write $M_k \coloneqq M_{k,0}$ and $Q_k \coloneqq Q_{k,0}$ for

[2]The topology of the network is available at https://github.com/bstabler/TransportationNetworks.



(a) Mean squared error of off-policy QMI (b) Exploitability of off-policy QMI



(c) Mean squared error of on-policy QMI (d) Exploitability of on-policy QMI

Figure 8: Performance comparison of different sample compensation factors on Sioux Falls network routing. As a baseline, the performance of FPI after 50 iterations is plotted as dashed lines.

notational simplicity. And when we restrict our discussion within an outer iteration, we omit the superscript k and write $M_t \coloneqq M_{k,t}$ and $Q_t \coloneqq Q_{k,t}$.

D Derivation of Online Stochastic Update for Population Distribution

We consider the squared L_2 error w.r.t. the MFNE (q^*, μ^*):

$$\frac{1}{2}\|M - \mu^*\|_2^2 = \frac{1}{2}\sum_{s\in\mathcal{S}}(M(s) - \mu^*(s))^2,$$

whose gradient w.r.t. M is

$$g_M^{(\text{real})} = M - \mu^* \overset{(2)}{=} M - \mathcal{P}_{q^*}\mu^* \overset{(6)}{=} M - \mathbb{E}_{q^*,\mu^*}[\delta_{s'}].$$

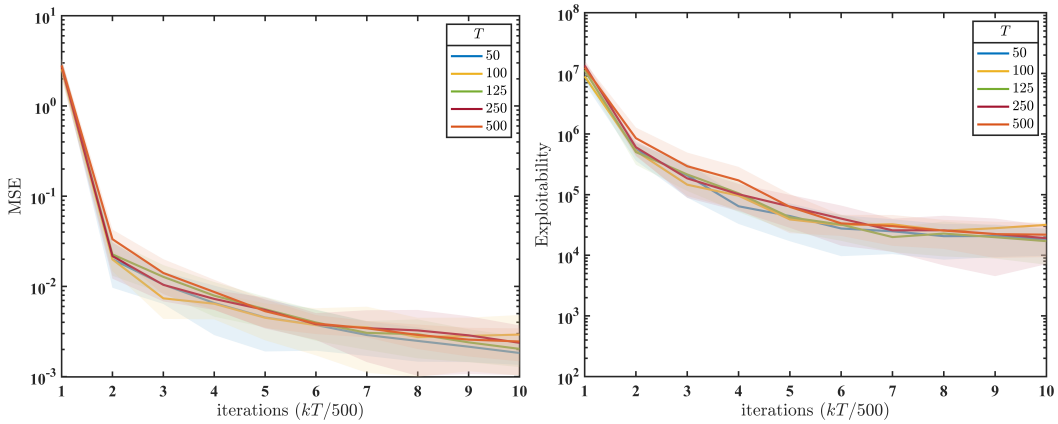
Since the information about MFNE is unavailable, we substitute it with the current policy-population pair (Q, M) (bootstrapping), giving the following *semi-gradient*:

$$g_M^{(\text{semi})} \coloneqq M - \mathbb{E}_{Q,M}[\delta_{s'}].$$

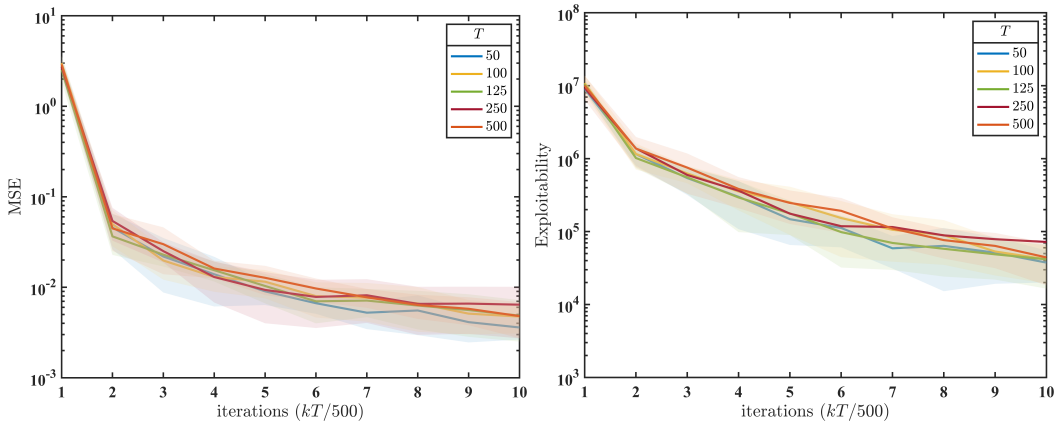
Therefore, a single online agent can update the population distribution estimate using the stochastic semi-gradient:

$$g_M(s') \coloneqq M - \delta_{s'},$$

which gives the M-value function update rule in (7). The Q-value function stochastic update rule can be derived similarly (see Sutton and Barto (2018)).



(a) Mean squared error of off-policy QMI (b) Exploitability of off-policy QMI



(c) Mean squared error of on-policy QMI (d) Exploitability of on-policy QMI

Figure 9: Performance comparison of different number of inner iterations given a fixed total sample size: $KT = 5000$ on Sioux Falls network routing.

E Preliminary Lemmas

This section provides two preliminary lemmas to assist the subsequent analysis.

Lemma 4 (Steady distribution difference (Mitrophanov, 2005, Corollary 3.1))**.** *For any two Q-value functions $Q_1, Q_2 \in \mathbb{R}^{S\times A}$, the difference between their induced steady distributions is bounded by*

$$\|\mu_{Q_1} - \mu_{Q_2}\|_{\mathrm{TV}} \le \sigma \|P_{Q_1} - P_{Q_2}\|_{\mathrm{TV}},$$

where we denote $P_Q \coloneqq P_{\Gamma_\pi(Q)}$, and

$$\|P_Q\|_{\mathrm{TV}} \coloneqq \sup_{\|q\|_{\mathrm{TV}}=1} \left\| \sum_{s\in\mathcal{S}} q(s) P_Q(s,\cdot) \right\|_{\mathrm{TV}}.$$

And the constant σ is defined as $\sigma = \hat{n} + m\rho^{\hat{n}}/(1-\rho)$, where $\hat{n} = \lceil \log_\rho m^{-1} \rceil$.

Furthermore, by Zou et al. (2019, Lemma 3), if the policy operator is Lipschitz continuous as specified in Assumption 4, we have

$$\|\mu_{Q_1} - \mu_{Q_2}\|_{\mathrm{TV}} \le L\sigma \|Q_1 - Q_2\|_2.$$

Lemma 5 (Young's inequality)**.** *For two points x, y in an inner product space, we have*

$$2\langle x, y\rangle \le \theta\|x\|^2 + 1/\theta\|y\|^2, \quad \forall\theta \in \mathbb{R},$$

where the norm is induced by the inner product. The above inequality further gives

$$\|x+y\|^2 \le (1+\theta)\|x\|^2 + (1+1/\theta)\|y\|^2, \quad \forall\theta \in \mathbb{R}.$$

Table 2: Notation.

Symbol	Definition	Reference
$\mathcal{S}, \mathcal{A}$	state and action space	Section 2.1
S, A	cardinality of $\mathcal{S}$ and $\mathcal{A}$	Section 2.1
r	reward function	Section 2.1
R	reward cap	Section 2.1
P	transition kernel	Section 2.1
(s, a, r, s', a')	online observation tuple	Section 3
γ	discount factor	Section 2.1
π	policy	Section 2.1
μ	population distribution	Section 2.1
$\Delta(\mathcal{S})$	probability simplex over $\mathcal{S}$	Section 2.1
π_M, Q_M	the BR w.r.t. $M \in \Delta(\mathcal{S})$	Section 2.1
μ_π, μ_Q	the IP w.r.t. π and $Q \in \mathbb{R}^{S\times A}$	Section 2.1
Q, M	Q-value and M-value function	Section 2.1
$\mathcal{T}_{Q,M}, \mathcal{P}_{Q,M}$	Bellman operator and transition operator	Definition 1
Γ_π	policy operator	Section 2.1
$\Gamma_{\mathrm{BR}}, \Gamma_{\mathrm{ID}}$	FPI operators	Definition 2
$\Gamma_Q, \Gamma_M, \hat{\Gamma}_{\mathrm{off}}, \hat{\Gamma}_{\mathrm{on}}$	QMI operators	Definition 3
κ	contraction parameter	Assumption 1
α, β	step sizes	Section 3
$\delta_{s'}$	indicator probability vector	Section 3
$m, \rho, \sigma, \lambda_{\min}$	MDP constants	Assumption 2
K, T	number of outer and inner iterations	Algorithm 1
L	Lipschitz constant for training smoothness	Assumptions 3 and 4
η	sample compensation factor	Appendix B
τ	backtracking period	Appendix G

F Analysis for Off-Policy QMI

F.1 Remark of Lemma 1

For Lemma 1, we utilize the result from Qu and Wierman (2020, Theorem 7), a finite time high probability L_∞ error bound for tabular Q-learning, which reads: suppose Assumption 2 holds and the step size is $\alpha_t = h/(t+t_0)$ where $h \geq 4/(\lambda_{\min}(1-\gamma))$ and $t_0 \geq \max\{4h, \lceil \log_2 2/\lambda_{\min}\rceil \log(4m)/\log \rho^{-1}\}$;[3]then with probability at least $1-\delta$,

$$\|\Gamma_Q(T)M - \Gamma_{\mathrm{BR}}M\|_\infty^2 = O\left(\frac{R^2 \log(T/\delta)}{\lambda_{\min}^2(1-\gamma)^5 T}\right),$$

where we omit the logarithmic dependencies on S, A and $\lambda_{\min}$. A high probability bound is stronger than a mean squared error bound. To see this, we have

$$\mathbb{E}\,\|\Gamma_Q(T)M - \Gamma_{\mathrm{BR}}M\|_\infty^2 \leq (1-\delta)\cdot O\left(\frac{R^2(\log T + \log \delta^{-1})}{\lambda_{\min}^2(1-\gamma)^5 T}\right) + \delta\cdot(2R)^2.$$

Substituting δ with $O(\log T/(\lambda_{\min}(1-\gamma)^5 T))$ gives the mean squared error bound we desire. By the relationship between the L_∞ norm and L_2 norm, we get the result presented in Lemma 1:

$$\mathbb{E}\,\|\Gamma_Q(T)M - \Gamma_{\mathrm{BR}}M\|_2^2 = O\left(\frac{SAR^2 \log T}{\lambda_{\min}^2(1-\gamma)^5 T}\right). \tag{9}$$

We acknowledge that there are other finite time error bounds for Q-learning. For example, Bhandari et al. (2018) established a finite time mean squared L_2 error bound for Q-learning with linear function approximation for

[3]Qu and Wierman (2020) uses a general mix time constant t_{mix}, which is bounded by $\log(4m)/\log\rho^{-1}$ by Assumption 2.

optimal stopping problems, and Li et al. (2023) sharpened the bound in (9) by a factor of $\frac{1}{\lambda_{\min}(1-\gamma)}$ with a constant step size. Since improving the constant dependencies in the finite error bounds is not our focus, we utilize the result from Qu and Wierman (2020) as their setting is the closest to ours.

F.2 Remark of Lemma 2

To directly invoke the results from Łatuszyński et al. (2013) for Lemma 2, we need to recast Assumption 2 using the terminology of small sets: the state space is $(1-\rho)$-small (please refer to Meyn and Tweedie (2012, Chapter 5) for the definitions). By Meyn and Tweedie (2012, Theorem 16.2.4), a $(1-\rho)$-small state space implies Assumption 2. Then, for a step size of $\beta_t = 1/(t+1)$, Łatuszyński et al. (2013, Theorem 3.1) gives a finite time mean squared L_∞ error bound for stationary Markov chain Monte Carlo methods. Łatuszyński et al. (2013, Remark 4.3) claims that

$$\mathbb{E}|(\Gamma_M(T)Q)_s - (\Gamma_{\text{IP}}Q)_s|^2 = O\left(\frac{1}{(1-\rho)^2 T}\right),$$

where $(x)_s$ represents the sth element of vector s. The result in Lemma 2 follows by relating the L_∞ norm and L_2 norm.

F.3 Proof of Theorem 1 for Off-Policy QMI

Proof. In this proof, we omit the subscript for the L_2 norm. We first bound the difference between $\hat{\Gamma}_{\text{off}}$ and Γ. For any $M \in \Delta(\mathcal{S})$, we have the decomposition

$$\begin{aligned}
&\left\|\hat{\Gamma}_{\text{off}}(T)M - \Gamma M\right\| \\
=&\left\|\left(\Gamma_M(T) \circ \Gamma_Q(T)\right) M - (\Gamma_{\text{IP}} \circ \Gamma_{\text{BR}})M\right\| \\
\le&\left\|\left(\Gamma_M(T) \circ \Gamma_Q(T)\right) M - (\Gamma_{\text{IP}} \circ \Gamma_Q(T))\, M\right\| + \left\|(\Gamma_{\text{IP}} \circ \Gamma_Q(T))\, M - (\Gamma_{\text{IP}} \circ \Gamma_{\text{BR}})M\right\| \\
=&\left\|\Gamma_M(T)Q - \Gamma_{\text{IP}}Q\right\| + \left\|\mu_Q - \mu_{q_M}\right\|, \qquad (10)
\end{aligned}$$

where we denote $Q \coloneqq \Gamma_Q(T)M$, and recall that μ_Q is the IP w.r.t. policy $\Gamma_\pi(Q)$. By Lemma 4, we have

$$\|\mu_Q - \mu_{q_M}\|_2 \le \|\mu_Q - \mu_{q_M}\|_1 \le \sigma \|P_Q - P_{q_M}\|_{\text{TV}}.$$

Then, by Assumption 3, we have $\|P_{Q_1} - P_{q_M}\|_{\text{TV}} \le L\|Q - q_M\|$. Plugging the above bound back into (10) gives

$$\left\|\hat{\Gamma}_{\text{off}}(T)M - \Gamma M\right\| \le \|\Gamma_M(T)Q - \Gamma_{\text{IP}}Q\| + L\sigma\|\Gamma_Q(T)M - \Gamma_{\text{BR}}M\|.$$

Then, by Lemmas 1, 2 and 5, we get

$$\begin{aligned}
\mathbb{E}\left\|\hat{\Gamma}_{\text{off}}(T)M - \Gamma M\right\|^2 \le& 2\mathbb{E}\left\|\Gamma_M(T)Q - \Gamma_{\text{IP}}Q\right\|^2 + 2(L\sigma)^2\mathbb{E}\left\|\Gamma_Q(T)M - \Gamma_{\text{BR}}M\right\|^2 \\
=& O\left(\frac{SA}{(1-\rho)^2 T}\right) + (L\sigma)^2 O\left(\frac{SAR^2\log T}{\lambda_{\min}^2(1-\gamma)^5 T}\right) \\
=& O\left(\frac{SAR^2L^2\sigma^2\log T}{\lambda_{\min}^2(1-\gamma)^5 T}\right), \qquad (11)
\end{aligned}$$

where the asymptotic notation holds when T is large enough such that $\frac{R^2L^2\sigma^2\log T}{\lambda_{\min}^2(1-\gamma)^5} \gg (1-\rho)^{-2}$.

We now bound the mean squared error of the M-value function after K outer iterations. Without loss of generality, we assume K is even. In this proof, we write $M_k = M_{k,0}$ for notational simplicity. By Lemma 5, we have

$$\begin{aligned}
\mathbb{E}\left\|\hat{\Gamma}_{\text{off}}^K M_0 - \mu^*\right\|^2 =& \mathbb{E}\left\|\hat{\Gamma}_{\text{off}} M_{K-2} - \Gamma\mu^*\right\|^2 \\
=& \mathbb{E}\left\|\hat{\Gamma}_{\text{off}} M_{K-2} - \Gamma M_{K-2} + \Gamma M_{K-2} - \Gamma\mu^*\right\|^2 \\
\le& (1+\kappa)\mathbb{E}\left\|\Gamma M_{K-2} - \Gamma\mu^*\right\|^2 + (1+1/\kappa)\mathbb{E}\left\|\hat{\Gamma}_{\text{off}} M_{K-2} - \Gamma M_{K-2}\right\|^2. \qquad (12)
\end{aligned}$$

By Assumption 1 and (11), we get

$$\begin{aligned}\mathbb{E}\left\|\hat{\Gamma}_{\text{off}}^{K}M_0-\mu^*\right\|^2 \leq&(1+\kappa)(1-\kappa)^2\mathbb{E}\left\|M_{K-2}-\mu^*\right\|^2+(1+1/\kappa)O\left(\frac{L^2SAR^2\sigma^2\log T}{\lambda_{\min}^2(1-\gamma)^5T}\right)\\ \leq&(1-\kappa)\mathbb{E}\left\|M_{K-2}-\mu^*\right\|^2+O\left(\frac{SAR^2L^2\sigma^2\log T}{\kappa\lambda_{\min}^2(1-\gamma)^5T}\right).\end{aligned}$$

Recursively applying the above inequality gives

$$\begin{aligned}\mathbb{E}\left\|\hat{\Gamma}_{\text{off}}^{K}M_0-\mu^*\right\|^2 \leq&(1-\kappa)^{K/2}\mathbb{E}\left\|M_0-\mu^*\right\|^2+O\left(\frac{SAR^2L^2\sigma^2\log T}{\kappa\lambda_{\min}^2(1-\gamma)^5T}\right)\sum_{k=1}^{K/2}(1-\kappa)^k\\ =&O\left(\exp\left(-\frac{\kappa K}{2}\right)+\frac{SAR^2L^2\sigma^2\log T}{\kappa^2\lambda_{\min}^2(1-\gamma)^5T}\right).\end{aligned}$$

□

G Analysis for On-Policy QMI with Population-Dependent Transition Kernel

G.1 Population-Dependent Transition Kernel

In this subsection, we generalize our setting to incorporate the influence of the population on agents' state transitions. Specifically, the transition kernel $P(s' \mid s, a, \mu)$ represents the probability of an agent transitioning to state s' when it takes action a at state s with the "current" population distribution being μ. When both the policy π and population distribution μ are fixed, we denote the transition kernel by $P_{\pi,\mu}$, which makes an agent's trajectory a stationary Markov chain, and $P_{\pi,\mu}(s, s')$ represents the probability of an agent transitioning to state s' from s when the dynamics are determined by π and μ.

This generalization necessitate several modifications in our setup: in (1), the expectation $\mathbb{E}_{\pi,\mu}$ is taken w.r.t. the transition kernel $P_{\pi,\mu}$; and the optimality equations (3) become

$$\begin{cases} Q &= \mathcal{T}_{Q,M}Q, \\ M &= \mathcal{P}_{Q,M}M, \end{cases}$$

where the Bellman operator now integrates w.r.t. the transition kernel $P_{Q,M}$:

$$\mathcal{T}_{Q,M}Q(s,a) = \mathbb{E}_{Q,M}[r(s,a,M)+\gamma Q(s',a')],$$

and the transition operator becomes

$$\mathcal{P}_{Q,M}M(s') = \sum_{s\in\mathcal{S}} P_{Q,M}(s,s')M(s),$$

where we denote $P_{Q,M} \coloneqq P_{\Gamma_\pi(Q),M}$.

Since now the transition operator also depends on the population, we need to redefine the IP operator in Definition 2 as

$$\Gamma_{\text{IP}} : \mathbb{R}^{S\times A}\times\Delta(\mathcal{S}) \to \Delta(\mathcal{S}),\ (Q,M)\mapsto \mu_{Q,M},$$

which returns the unique fixed point of the transition operator $\mathcal{P}_{Q,M}$ for any value function Q and population distribution M. To composite the BR operator and IP operator, we need to extend the BR operator:

$$\Gamma_{\text{BR}} : \Delta(\mathcal{S}) \to \mathbb{R}^{S\times A}\times\Delta(\mathcal{S}),\ M\mapsto (Q_M, M),$$

i.e., keep a copy of the original population while calculating the BR. The FPI operator is still the composition of the two operators: $\Gamma = \Gamma_{\text{IP}}\circ\Gamma_{\text{BR}}$.

Although the definition of the transition operator $\mathcal{P}_{Q,M}$ has changed, the same derivation in Section 3 yields the same online stochastic update rule for the population distribution specified in (7). Consequently, the only modification required in Algorithm 1 is that, within the kth outer iteration, we fix the reference population distribution as $M_{k,0}$ in the transition kernel $P_{\pi_{k,t},M_{k,0}}$, similar to that we fix it in the reward function $r(s_t, a_t, M_{k,0})$.

G.2 Proof of Lemma 3

In this subsection, we prove Lemma 3 considering general population-dependent transition kernels, which contain population-independent transition kernels as special cases. Note that the on-policy QMI operator is defined within a single outer iteration, so we omit all the subscripts related to the outer iteration. We make the following notational simplification:

$$M_t \coloneqq \hat{\Gamma}_{\text{on}}(t)M, \quad \mu^\dagger \coloneqq \Gamma_{\text{IP}}(q^\dagger, M) \coloneqq \Gamma_{\text{IP}}(\Gamma_{\text{BR}}M) = \Gamma M, \quad \Delta_t \coloneqq M_t - \mu^\dagger.$$

Also, although $\hat{\Gamma}_{\text{on}}$ does not directly return the Q-value function, it will update the Q-value function, and thus the behavior policy, along the process. We denote $\hat{Q}_t$ the *mixed* Q-value function at inner time step t (see Algorithm 1), and $\mu_t = \Gamma_{\text{IP}}(\hat{Q}_t, M)$ the population distribution induced by the transition kernel $P_{\hat{Q}_t, M}$.

Since the on-policy QMI trajectories are not stationary anymore, we need a special virtual *backtracking* process to assist our analysis, which enables us to use the mixing property of stationary Markov chains (Zou et al., 2019). Specifically, we consider a virtual trajectory, where we backtrack a period τ and fix the behavior policy as $\Gamma_\pi(\hat{Q}_{t-\tau})$ after time step $t-\tau$. Thus, this virtual trajectory is stationary after time step $t-\tau$, and its observation distribution will rapidly converge to the steady distribution w.r.t. $\hat{Q}_{t-\tau}$ due to the mixing property of stationary MDPs (Assumption 2). We denote $\tilde{\delta}_t$ the indicator function/vector such that $\tilde{\delta}_t(s) = \mathbb{1}(s = \tilde{s}_t)$, where $\tilde{s}_t$ is the virtual state observation on this virtual trajectory at time step t. For the actual state s_t at time step t induced by the actual nonstationary trajectory, we denote $\delta_t \coloneqq \delta_{s_t}$. In this section, we omit the subscript for the L_2 norm.

To prove Lemma 3, we first decompose the difference Δ_t recursively.

Lemma 6 (Error decomposition)**.**

$$\begin{aligned}
\mathbb{E}\left\|\Delta_{t+1}\right\|^2 \leq & (1-2\beta_t)\mathbb{E}\left\|\Delta_t\right\|^2 + 4\beta_t^2 \\
& + 2\beta_t \mathbb{E}\langle \Delta_t, \mu_t - \mu^\dagger \rangle && \text{(control)} \\
& + 2\beta_t \mathbb{E}\langle \Delta_t, \mu_{t-\tau} - \mu_t \rangle && \text{(progress)} \\
& + 2\beta_t \mathbb{E}\langle \Delta_t, \tilde{\delta}_{t+1} - \mu_{t-\tau} \rangle && \text{(mixing)} \\
& + 2\beta_t \mathbb{E}\langle \Delta_t, \delta_{t+1} - \tilde{\delta}_{t+1} \rangle. && \text{(backtracking)}
\end{aligned}$$

Proof. By the update rule (7), we have

$$\mathbb{E}\left\|\Delta_{t+1}\right\|^2 = \mathbb{E}\left\|M_t + \beta_t(\delta_{t+1} - M_t) - \mu^\dagger\right\|^2 = \mathbb{E}\|\Delta_t\|^2 + 2\beta_t \mathbb{E}\left\langle \Delta_t, \delta_{t+1} - M_t \right\rangle + \beta_t^2 \mathbb{E}\|\delta_{t+1} - M_t\|^2.$$

Since δ_{t+1} and M_t are both probability vectors, we have $\|\delta_{t+1} - M_t\| \leq \|\delta_{t+1}\| + \|M_t\| \leq \|\delta_{t+1}\|_1 + \|M_t\|_1 = 2$. Then, we apply the following decomposition:

$$\delta_{t+1} - M_t = -(M_t - \mu^\dagger) + (\mu_t - \mu^\dagger) + (\mu_{t-\tau} - \mu_t) + (\tilde{\delta}_{t+1} - \mu_{t-\tau}) + (\delta_{t+1} - \tilde{\delta}_{t+1}),$$

which gives the result. □

We next provide four lemmas, each bounding one term in the above decomposition.

Lemma 7 (Control)**.** *With a step size of $\alpha_t = \frac{b}{\lambda_{\min}(1-\gamma)(t+1+a)}$ for the Q-value function update, where a and b are constants ensuring that the initial step size α_0 is small enough (see Anonymous (2023)), we have*

$$\mathbb{E}\left\|\mu_t - \mu^\dagger\right\|^2 = O\left(\frac{SAR^2L^2\sigma^2 \log t}{\lambda_{\min}^2(1-\gamma)^4 t}\right).$$

Proof. First, by Lemma 4, we can bound the distribution difference by the control (Q-value function) difference:

$$\mathbb{E}\left\|\mu_t - \mu^\dagger\right\|^2 \leq (L\sigma)^2 \mathbb{E}\left\|\hat{Q}_t - q^\dagger\right\|^2.$$

Then, by Anonymous (2023, Corollary 2.2), we have

$$\mathbb{E}\left\|\hat{Q}_t - q^\dagger\right\|^2 = O\left(\frac{SAR^2 \log t}{\lambda_{\min}^2(1-\gamma)^4 t}\right).$$

Plugging this bound back gives the result. □

Lemma 8 (Progress). *Given the policy update rule in Algorithm 1, we have*

$$\mathbb{E}\left\|\mu_t - \mu_{t-\tau}\right\|^2 = O\left(\frac{R^2L^2\sigma^2\tau^2}{(1-\gamma)^2(t-\tau)^2}\right).$$

Proof. First, by Lemma 4, we can bound the distribution progress by the Q-value function progress:

$$\mathbb{E}\left\|\mu_t - \mu_{t-\tau}\right\|^2 \le (L\sigma)^2\mathbb{E}\left\|\hat{Q}_t - \hat{Q}_{t-\tau}\right\|^2.$$

We first bound the one-step mixed Q-value function progress. According to the convex combination in Anonymous (2023, Corollary 2.2), $w_t = t + a$ and $W_t = \sum_{l=0}^{t} w_l \asymp t^2$. We have

$$\begin{aligned}
\left\|\hat{Q}_{t+1} - \hat{Q}_t\right\| =& \left\|\frac{w_{t+1}}{W_{t+1}}Q_{t+1} + \frac{W_t}{W_{t+1}}\hat{Q}_t - \hat{Q}_t\right\| \\
\le& \frac{w_{t+1}}{W_{t+1}}\left\|Q_{t+1}\right\| + \left|\frac{W_t}{W_{t+1}} - 1\right|\left\|\hat{Q}_t\right\| \\
\le& \frac{R}{1-\gamma}\cdot O\left(1/t\right) + \frac{R}{1-\gamma}\cdot O\left(1/t\right) \\
=& O\left(\frac{R}{(1-\gamma)t}\right),
\end{aligned}$$

where we use the fact that the Q-value function is uniformly bounded by $R/(1-\gamma)$ and $W_t \asymp t^2$. Then, by the triangle inequality, we have

$$\left\|\hat{Q}_t - \hat{Q}_{t-\tau}\right\| = O\left(\frac{R\tau}{(1-\gamma)(t-\tau)}\right).$$

Plugging this bound back gives the result. □

Lemma 9 (Mixing). *Let τ be the backtracking period. Then, for any $\theta > 0$, we have*

$$\left|2\mathbb{E}\left\langle \Delta_t, \tilde{\delta}_{t+1} - \mu_{t-\tau}\right\rangle\right| \le \theta\mathbb{E}\|\Delta_t\|^2 + m^2\rho^{2\tau}/\theta.$$

Proof. We first bound the difference between $\tilde{\delta}_{t+1}$ and $\mu_{t-\tau}$ conditioned on the filtration $\mathcal{F}_{t-\tau}$ containing all the randomness before time step $t-\tau$. Recall that $\mathbb{E}[\tilde{\delta}_{t+1}] = \Pr(\tilde{s}_{t+1} = \cdot) \in \Delta(\mathcal{S})$, which gives

$$\left\|\mathbb{E}\left[\tilde{\delta}_{t+1} - \mu_{t-\tau}\,\middle|\,\mathcal{F}_{t-\tau}\right]\right\| = \left\|\mathbb{E}\left[\tilde{\delta}_{t+1}\,\middle|\,\mathcal{F}_{t-\tau}\right] - \mu_{t-\tau}\right\| = \left\|\Pr(\tilde{s}_{t+1} = \cdot\,|\,\mathcal{F}_{t-\tau}) - \mu_{t-\tau}\right\|. \tag{13}$$

Note that $\tilde{s}_{t+1}$ is on the virtual stationary trajectory following a fixed policy $\Gamma_\pi(\hat{Q}_{t-\tau})$. Thus, Assumption 2 gives

$$\left\|\mathbb{E}\left[\tilde{\delta}_{t+1} - \mu_{t-\tau}\,\middle|\,\mathcal{F}_{t-\tau}\right]\right\| \le \left\|\Pr(\tilde{s}_{t+1} = \cdot\,|\,\mathcal{F}_{t-\tau}) - \mu_{t-\tau}\right\|_{\mathrm{TV}} \le m\rho^\tau.$$

Also note that conditioned on $\mathcal{F}_{t-\tau}$, the virtual trajectory and the actual one are independent. Therefore, we have

$$\begin{aligned}
\left|\mathbb{E}\left\langle \Delta_t, \tilde{\delta}_{t+1} - \mu_{t-\tau}\right\rangle\right| =& \left|\mathbb{E}\left[\mathbb{E}\left[\left\langle \Delta_t, \tilde{\delta}_{t+1} - \mu_{t-\tau}\right\rangle\,\middle|\,\mathcal{F}_{t-\tau}\right]\right]\right| \\
=& \left|\mathbb{E}\left\langle \mathbb{E}\left[\Delta_t\,|\,\mathcal{F}_{t-\tau}\right], \mathbb{E}\left[\tilde{\delta}_{t+1} - \mu_{t-\tau}\,\middle|\,\mathcal{F}_{t-\tau}\right]\right\rangle\right| \\
\le& \mathbb{E}\left[\left\|\mathbb{E}\left[\Delta_t\,|\,\mathcal{F}_{t-\tau}\right]\right\|\left\|\mathbb{E}\left[\tilde{\delta}_{t+1} - \mu_{t-\tau}\,\middle|\,\mathcal{F}_{t-\tau}\right]\right\|\right] \\
\le& \mathbb{E}\|\Delta_t\|\cdot m\rho^\tau.
\end{aligned}$$

Finally, invoking Lemma 5 gives the result. □

Lemma 10 (Backtracking). *Let τ be the backtracking period. Suppose the step size is non-increasing. Then, for any $\theta > 0$, we have*

$$\left|2\mathbb{E}\left\langle \Delta_t, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle\right| \leq \theta\mathbb{E}\|\Delta_t\|^2 + 4\beta_{t-\tau}\tau + O\left(\frac{LR\tau^3}{(1-\gamma)(t-\tau)}\left(\beta_{t-\tau} + \frac{LR\tau}{\theta(1-\gamma)(t-\tau)}\right)\right).$$

Proof. Similar to (13), we have

$$\left\|\mathbb{E}\left[\delta_{t+1} - \tilde{\delta}_{t+1} \,\middle|\, \mathcal{F}_{t-\tau}\right]\right\| = \left\|\Pr\left(s_{t+1} = \cdot \,|\, \mathcal{F}_{t-\tau}\right) - \Pr\left(\tilde{s}_{t+1} = \cdot \,|\, \mathcal{F}_{t-\tau}\right)\right\|.$$

By Zou et al. (2019, Equation 46), we have

$$\left\|\Pr\left(s_{t+1} = \cdot \,|\, \mathcal{F}_{t-\tau}\right) - \Pr\left(\tilde{s}_{t+1} = \cdot \,|\, \mathcal{F}_{t-\tau}\right)\right\| \leq L\sum_{l=t-\tau}^{t}\mathbb{E}\|\hat{Q}_l - \hat{Q}_{t-\tau}\| = O\left(\frac{LR\tau^2}{(1-\gamma)(t-\tau)}\right). \tag{14}$$

Here, since δ_{t+1} and M_t are correlated conditioned on $\mathcal{F}_{t-\tau}$, we cannot directly get the result analogous to that in Lemma 9. We apply the following decomposition:

$$\left|2\mathbb{E}\left\langle \Delta_t, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle\right| \leq \underbrace{\left|2\mathbb{E}\left\langle M_t - M_{t-\tau}, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle\right|}_{H_1} + \underbrace{\left|2\mathbb{E}\left\langle M_{t-\tau} - \mu^\dagger, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle\right|}_{H_2}.$$

For H_1, by Lemma 5, we have

$$\begin{aligned}
H_1 \leq& \theta'\mathbb{E}\|M_t - M_{t-\tau}\|^2 + 1/\theta'\mathbb{E}\|\delta_{t+1} - \tilde{\delta}_{t+1}\|^2 \\
\leq& \theta'\mathbb{E}\left\|\sum_{l=t-\tau}^{t-1}\beta_l g_{M_l}\right\|^2 + 1/\theta' \cdot 4 \\
\leq& 4\theta'\beta_{t-\tau}^2\tau^2 + 4/\theta',
\end{aligned} \tag{15}$$

where we use the fact that $\|M_1 - M_2\|_2 \leq 2$ for any $M_1, M_2 \in \Delta(\mathcal{S})$. Let $\theta' = 1/(\beta_{t-\tau}\tau)$. We get

$$H_1 \leq 4\beta_{t-\tau}\tau.$$

For H_2, we can apply (14), which gives

$$\begin{aligned}
H_2 =& 2\left|\mathbb{E}\left[\mathbb{E}\left[\left\langle \Delta_{t-\tau}, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle \,\middle|\, \mathcal{F}_{t-\tau}\right]\right]\right| \\
=& 2\left|\mathbb{E}\left\langle \mathbb{E}\left[\Delta_{t-\tau} \,|\, \mathcal{F}_{t-\tau}\right], \mathbb{E}\left[\delta_{t+1} - \tilde{\delta}_{t+1} \,\middle|\, \mathcal{F}_{t-\tau}\right]\right\rangle\right| \\
\leq& 2\mathbb{E}\left[\left\|\mathbb{E}\left[\Delta_{t-\tau} \,|\, \mathcal{F}_{t-\tau}\right]\right\|\left\|\mathbb{E}\left[\delta_{t+1} - \tilde{\delta}_{t+1} \,\middle|\, \mathcal{F}_{t-\tau}\right]\right\|\right] \\
=& \mathbb{E}\|\Delta_{t-\tau}\| \cdot O\left(\frac{LR\tau^2}{(1-\gamma)(t-\tau)}\right).
\end{aligned}$$

Similar to (15), we have

$$\mathbb{E}\|\Delta_{t-\tau}\| \leq \mathbb{E}\|\Delta_t\| + \mathbb{E}\|M_t - M_{t-\tau}\| \leq \mathbb{E}\|\Delta_t\| + 2\beta_{t-\tau}\tau.$$

Plugging the above bounds on H_1 and H_2 back gives

$$\left|2\mathbb{E}\left\langle \Delta_t, \delta_{t+1} - \tilde{\delta}_{t+1}\right\rangle\right| \leq 4\tau\beta_{t-\tau} + (\mathbb{E}\|\Delta_t\| + 2\beta_{t-\tau}\tau) \cdot O\left(\frac{LR\tau^2}{(1-\gamma)(t-\tau)}\right).$$

Applying Lemma 5 on $\mathbb{E}\|\Delta_t\| \cdot O\left(\frac{LR\tau^2}{(1-\gamma)(t-\tau)}\right)$ gives the result. □

Given the above five lemmas, we are ready to prove Lemma 3.

Proof of Lemma 3. We first plug Lemmas 7 to 10 into Lemma 6:

$$\begin{aligned}
\mathbb{E}\left\|\Delta_{t+1}\right\|^2 \leq & (1-2\beta_t)\mathbb{E}\left\|\Delta_t\right\|^2 + 4\beta_t^2 \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + \beta_t/\theta\mathbb{E}\|\mu_t - \mu^\dagger\|^2 \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + \beta_t/\theta\mathbb{E}\|\mu_{t-\tau} - \mu_t\|^2 \\
& + 2\beta_t\left|\mathbb{E}\langle\Delta_t, \tilde{\delta}_{t+1} - \mu_{t-\tau}\rangle\right| \\
& + 2\beta_t\left|\mathbb{E}\langle\Delta_t, \delta_{t+1} - \tilde{\delta}_{t+1}\rangle\right| \\
\leq & (1-2\beta_t)\mathbb{E}\left\|\Delta_t\right\|^2 + 4\beta_t^2 \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + \frac{\beta_t}{\theta}\cdot O\left(\frac{SAR^2L^2\sigma^2\log t}{\lambda_{\min}^2(1-\gamma)^4 t}\right) && \text{(Lemma 7)} \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + \frac{\beta_t}{\theta}\cdot O\left(\frac{R^2L^2\sigma^2\tau^2}{(1-\gamma)^2(t-\tau)^2}\right) && \text{(Lemma 8)} \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + \frac{\beta_t}{\theta}\cdot m^2\rho^{2\tau} && \text{(Lemma 9)} \\
& + \beta_t\theta\mathbb{E}\|\Delta_t\|^2 + 4\beta_{t-\tau}\beta_t\tau + \beta_t\cdot O\left(\frac{LR\tau^3}{(1-\gamma)(t-\tau)}\left(\beta_{t-\tau} + \frac{LR\tau}{\theta(1-\gamma)(t-\tau)}\right)\right). && \text{(Lemma 10)}
\end{aligned}$$

After some arrangement, we get

$$\begin{aligned}
\mathbb{E}\|\Delta_{t+1}\|^2 \leq & (1-\beta_t(2-4\theta))\mathbb{E}\|\Delta_t\|^2 + 4\beta_t^2 + \frac{\beta_t}{\theta} \\
& \cdot O\left(\frac{SAR^2L^2\sigma^2\log t}{\lambda_{\min}^2(1-\gamma)^4 t} + \frac{R^2L^2\tau^4}{(1-\gamma)^2(t-\tau)^2} + \frac{\theta LR\beta_{t-\tau}\tau^3}{(1-\gamma)(t-\tau)} + \frac{R^2L^2\sigma^2\tau^2}{(1-\gamma)^2(t-\tau)^2} + \beta_{t-\tau}\tau + m^2\rho^{2\tau}\right).
\end{aligned} \tag{16}$$

Note that we have not yet set θ and the backtracking period τ. Let $\theta = 1/4$ and $\tau = \lceil \log\frac{\beta_t}{m}/\log\rho \rceil$. Recall that $\beta_t \asymp 1/t$ and $\alpha_t \asymp 1/(\lambda_{\min}(1-\gamma)t)$; then $\tau \asymp \log t$, $\beta_t \asymp \beta_{t-\tau}$, and $\alpha_t \asymp \alpha_{t-\tau}$. Therefore, (16) gives

$$\begin{aligned}
& \mathbb{E}\|\Delta_{t+1}\|^2 \\
\leq & (1-\beta_t)\mathbb{E}\|\Delta_t\|^2 + 4\beta_t^2 + \beta_t\cdot O\left(\frac{SAR^2L^2\sigma^2\log t}{\lambda_{\min}^2(1-\gamma)^4 t} + \frac{R^2L^2\log^4 t}{(1-\gamma)^2t^2} + \frac{RL\log^3 t}{(1-\gamma)t^2} + \frac{R^2L^2\sigma^2\log^2 t}{(1-\gamma)^2t^2} + \frac{\log t}{t} + \frac{1}{t^2}\right) \\
= & (1-\beta_t)\mathbb{E}\|\Delta_t\|^2 + \beta_t\cdot O\left(\frac{SAR^2L^2\sigma^2\log t}{\lambda_{\min}^2(1-\gamma)^4 t}\right),
\end{aligned} \tag{17}$$

where the asymptotic notation in the last equality holds when $t \gg 1$ and $1 = O(SAR^2L^2\sigma^2/(\lambda_{\min}(1-\gamma)^4))$.

Recall our step size configuration: $\beta_t = 1/(t+1)$. Dividing both sides of (17) by β_t gives

$$(t+1)\mathbb{E}\|\Delta_{t+1}\|^2 \leq t\mathbb{E}\|\Delta_t\|^2 + C\cdot O\left(\frac{\log t}{t}\right),$$

where $C \coloneqq SAR^2L^2\sigma^2/(\lambda_{\min}^2(1-\gamma)^4)$.[4] Then, we get

$$\begin{aligned}
T\mathbb{E}\|\Delta_T\|^2 \leq & (T-1)\mathbb{E}\|\Delta_{T-1}\|^2 + C\cdot O\left(\frac{\log(T-1)}{T-1}\right) \\
\leq & C\sum_{t=1}^{T} O\left(\frac{\log t}{t}\right) \\
= & C\cdot O(\log T),
\end{aligned}$$

where the last equality is by Merten's theorem. And the result follows. □

[4] Here, with a slight abuse of notation, C is smaller than the one in Theorem 1 by a factor of $1/(1-\gamma)$. The larger C in Theorem 1 accounts for the incorporation of Lemma 1. Actually, the dependencies in Lemma 1 can be improved to match the ones in Lemma 3; see Li et al. (2023).

G.3 Proof of Theorem 1 for On-Policy QMI

Proof. Similar to (12), by Lemma 5, we get

$$\mathbb{E}\left\|\hat{\Gamma}_{\text{on}}^{K} M_0 - \mu^*\right\|^2 \leq (1+\kappa)\mathbb{E}\left\|\Gamma M_{K-1} - \Gamma\mu^*\right\|^2 + (1+1/\kappa)\mathbb{E}\left\|\hat{\Gamma}_{\text{on}} M_{K-1} - \Gamma M_{K-1}\right\|^2.$$

By Assumption 1 and Lemma 3, we get

$$\begin{aligned}
\mathbb{E}\left\|\hat{\Gamma}_{\text{on}}^{K} M_0 - \mu^*\right\|^2 \leq & (1+\kappa)(1-\kappa)^2\mathbb{E}\left\|M_{K-1} - \mu^*\right\|^2 + (1+1/\kappa)C \cdot O\left(\frac{\log T}{T}\right) \\
\leq & (1-\kappa)\mathbb{E}\left\|M_{K-1} - \mu^*\right\|^2 + C \cdot O\left(\frac{\log T}{\kappa T}\right) \\
\leq & (1-\kappa)^K\left\|M_0 - \mu^*\right\|^2 + \sum_{k=1}^{K}(1-\kappa)^k C \cdot O\left(\frac{\log T}{T}\right) \\
= & O\left(\exp(-\kappa K) + \frac{SAR^2L^2\sigma^2\log T}{\kappa^2\lambda_{\min}(1-\gamma)^4 T}\right).
\end{aligned}$$

□

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   Yes. Please see Sections 2 and 4.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Yes. Please see Section 5.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes. All the implementation details are provided in Appendix B. An implementation in MATLAB is provided in the supplementary materials.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Yes. Please see the lemmas and theorems in the Section 5.
   (b) Complete proofs of all theoretical results.
   Yes. Please see Appendices F and G.
   (c) Clear explanations of any assumptions.
   Yes. All assumptions are followed by clear explanations and justifications, supported by relevant references.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   Yes. Please see Appendix B.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Yes. Please see Appendix B.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   Yes. Please see Section 6.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
   Not Applicable. The implementation of our experiments and related algorithms are simple and straightforward. The numerical results can be reproduced on any device, without requiring GPU, internal clusters, or cloud services. Moreover, our results are unaffected by hardware specifics as they do not involve hardware-dependent metrics such as computation time and memory size.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets.
   Not Applicable.
   (b) The license information of the assets, if applicable.
   Not Applicable.
   (c) New assets either in the supplemental material or as a URL, if applicable.
   Not Applicable.
   (d) Information about consent from data providers/curators.
   Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
   Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots.
   Not Applicable.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
Not Applicable.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
Not Applicable.