Transition-Informed Reinforcement Learning for Large-Scale Stackelberg Mean-Field Games

Pengdeng Li[1], Runsheng Yu[2], Xinrun Wang[1]*, Bo An[1]

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]Hong Kong University of Science and Technology, Hong Kong, China
{pengdeng.li, xinrun.wang, boan}@ntu.edu.sg, runshengyu@gmail.com

Abstract

Many real-world scenarios including fleet management and Ad auctions can be modeled as Stackelberg mean-field games (SMFGs) where a leader aims to incentivize a large number of homogeneous self-interested followers to maximize her utility. Existing works focus on cases with a small number of heterogeneous followers, e.g., 5-10, and suffer from scalability issue when the number of followers increases. There are three major challenges in solving large-scale SMFGs: i) classical methods based on solving differential equations fail as they require exact dynamics parameters, ii) learning by interacting with environment is data-inefficient, and iii) complex interaction between the leader and followers makes the learning performance unstable. We address these challenges through transition-informed reinforcement learning. Our main contributions are threefold: i) we first propose an RL framework, the Stackelberg mean-field update, to learn the leader's policy without priors of the environment, ii) to improve the data efficiency and accelerate the learning process, we then propose the Transition-Informed Reinforcement Learning (TIRL) by leveraging the instantiated empirical Fokker-Planck equation, and iii) we develop a regularized TIRL by employing various regularizers to alleviate the sensitivity of the learning performance to the initialization of the leader's policy. Extensive experiments on fleet management and food gathering demonstrate that our approach can scale up to 100,000 followers and significantly outperform existing baselines.

1 Introduction

Learning to incentivize a large population of homogeneous self-interested followers is of great importance for extensive real-world problems. For example, in the e-hailing driver repositioning (EDRP) (Shou and Di 2020), to improve order response rate, the platform incentivizes enormous drivers to spread across different areas of the city by taking service charge. Other scenarios include Ad auctions (Guo et al. 2019), electronic toll collection (Qiu, Chen, and An 2019), and mechanism design for e-commerce (Cai et al. 2018). These scenarios can be modeled as large-scale SMFGs where a leader, e.g., the e-hailing or Ad-exchange platform, incentivizes enormous homogeneous self-interested followers, e.g., taxi drivers or advertisers, to maximize her utility.

There are several works related to the problem of incentivizing self-interested followers including mind-aware multi-agent management (M^3RL) (Shu and Tian 2019) and expensive coordination (Yu et al. 2020). Nevertheless, these works consider a small number of heterogeneous followers and suffer from the scalability issue when the number of followers increases. In contrast, Stackelberg mean-field game (SMFG) provides a powerful tool to model the scenarios with enormous homogeneous self-interested followers. SMFGs are established with consideration of various constraints and there are analytical results such as the existence and uniqueness or the explicit form of solution for special cases (e.g., linear-quadratic) (Bensoussan et al. 2017; Fu and Horst 2020; Huang and Yang 2020). However, these methods are infeasible in practice due to the following critical challenges. (1) Classical methods in the previous works typically involve solving a set of differential equations, which fail as they require exact dynamics parameters (e.g., the transition rate matrix) to derive closed-form solutions in special cases such as the linear-quadratic model. Moreover, none of the existing works have tried to solve the SMFGs by employing RL algorithms that can learn the policies without priors of the environments. (2) Learning by interacting with the environment is data-inefficient and thus, typically requires long learning time before achieving desirable performance. (3) The interaction between the leader and followers is complex, i.e., the leader takes actions under the consideration of the followers' rational responses while the followers adapt their policies given the leader's policy, which renders the learning performance unstable, i.e., different initializations of the leader's policy could result in dramatically different learning performance.

In this paper, we address the above challenges and provide the following three contributions. (1) We propose the first RL framework, the Stackelberg mean-field update (SMFU), to learn the leader's policy without priors of the environments. (2) As the SMFU is a model-free algorithm that is data-inefficient, which requires an excessive number of interactions with the environment and thus, takes long learning time to achieve desirable performance, we propose a novel learning framework, the Transition-Informed Reinforcement Learning (TIRL), to improve the data efficiency and in turn accelerate the learning process. Specifically, we first use the experience tuples generated by interacting with the environ-

[*] Corresponding author.

ment to learn the followers' transition function and instantiate the empirical Fokker-Planck (EFP) equation. Then, new experience tuples can be generated by leveraging the EFP equation without additional interactions with the environment. (3) To alleviate the sensitivity of the TIRL to the initialization of leader's policy, we develop a regularized TIRL by employing various regularizers to further improve its stability. Extensive experiments on fleet management and food gathering scenarios show that our approach can scale up to 100,000 followers and significantly outperform baselines.

2 Related Works

Our work is closely related to Stackelberg mean-field game and model-based reinforcement learning.

Stackelberg Mean-Field Games (SMFGs). Learning to incentivize self-interested followers has received increasing attention recently. The reinforcement mechanism design applies RL to mechanism design in e-commerce (Cai et al. 2018). M^{3}RL (Shu and Tian 2019) considers the optimization of the manager's strategy against rule-based followers. The setting is extended by (Yu et al. 2020), which proposes a Stackelberg Markov game to model the interaction between the leader and RL-based followers and abstraction-based algorithms to compute the leader's strategy. Nevertheless, these works consider a small number of heterogeneous followers and suffer from the scalability issue when the number of followers increases since the learning takes place in the product space of state and action spaces across followers such that the complexity of finding leader's optimal policy grows exponentially with the number of followers. In many real-world scenarios such as EDRP and Ad auctions, the leader needs to incentivize a large number of homogeneous followers, which can be modeled as SMFGs. In (Bensoussan et al. 2017), the SMFG is established and the optimal control of the leader is derived for the linear-quadratic case. The SMFG with terminal state constraints is considered in (Fu and Horst 2020), and dynamic programming is employed to determine the equilibrium strategy in (Huang and Yang 2020). In addition, the SMFG is applied to model the epidemic control problem in (Aurell et al. 2022). However, these works require exact dynamics parameters and hence, are infeasible in practice when no prior of environment is available. Moreover, in contrast to the mean-field game where RL has been widely adopted (Guo et al. 2019; Subramanian and Mahajan 2019), none of the existing works have tried to solve the SMFGs by employing RL algorithms. In this work, we propose the first RL framework which we call the Stackelberg mean-field update (SMFU) to learn the leader's policy without priors of environments.

Model-Based RL. MBPO (Janner et al. 2020) and MuZero (Schrittwieser et al. 2020) achieve the asymptotic performance similar to the state-of-the-art model-free algorithms. However, these works focus on single-agent environments which are contrary to our setting with the presence of multiple players. For multi-agent settings, in (Krupnik, Mordatch, and Tamar 2020), the MBRL with latent variable models is extended to multi-agent settings. In (Kamra et al. 2020), the interaction graph-based trajectory prediction methods are suggested. In (Zhang et al. 2021), a decentralized MBRL method is proposed with consideration of multiple opponent models. Unfortunately, these works either consider zero-sum games or treat the players as atomic and learn separate models for the players which could be computationally expensive. To overcome the difficulties, we propose Transition-Informed Reinforcement Learning (TIRL). The key insight is that, instead of predicting a single state for each follower which is atomic, TIRL directly computes the followers' new state distribution by using the learned transition function, which is, on the contrary, a non-atomic (Aumann and Shapley 2015) approach.

3 Stackelberg Mean-Field Game

In this section, we present the definition and an illustrative example of our game model.

3.1 Game Formulation

Let us start with the case of finite followers. Consider the leader-follower game played between one leader and N followers indexed by $\mathcal{N} = \{1, \cdots, N\}$. Let $\mathcal{X}/\mathcal{U}$ and $\mathcal{S}/\mathcal{A}$ denote the leader's and followers' state/action spaces, respectively, where $\mathcal{S}$ and $\mathcal{A}$ are finite and identical to the followers. Let $\mathcal{H} = \{0, 1, \cdots, H\}$ denote the time index set. At time $h \in \mathcal{H}$, the leader's and follower i's states are $x_h \in \mathcal{X}$ and $s_h^i \in \mathcal{S}$, respectively. Let $\mathbb{G}_h^N(k) = N^{-1} \sum_{i \in \mathcal{N}} \mathbf{1}_k(s_h^i)$ denote the empirical state distribution of the followers, where $\mathbf{1}_k(s_h^i)$ is an indicator function: $\forall k \in \mathcal{S}$, $\mathbf{1}_k(s_h^i) = 1$ if $s_h^i = k$ and 0 otherwise. Note that $\mathbb{G}_h^N \in \Delta(\mathcal{S})$, the space of probability measures on $\mathcal{S}$. The transitions of states are determined by continuous functions $P^L : \mathcal{X} \times \mathcal{U} \times \Delta(\mathcal{S}) \to \Delta(\mathcal{X})$ and $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$. Observe that P does not depend on the state distribution as is common in the literature (Lasry and Lions 2007; Perolat et al. 2022; Perrin et al. 2020). Consider for the leader and each follower i the Markov policies $\mu \in \Upsilon$ and $\pi^i \in \Pi$ where $\mu : \mathcal{X} \to \Delta(\mathcal{U})$ and $\pi^i : \mathcal{S} \to \Delta(\mathcal{A})$. Υ and Π are the spaces of the leader's and followers' all Markov policies, respectively, where Υ is assumed to be compact, i.e., closed and bounded. The leader takes an action $u_h \sim \mu$ and then, the follower i takes an action $a_h^i \sim \pi^i$. Their states change to $x_{h+1} \sim P^L$ and $s_{h+1}^i \sim P$, respectively. The reward functions of the leader and follower i are $r^L(x_h, u_h, \mathbb{G}_h^N)$ and $r^i(u_h, s_h^i, a_h^i, \mathbb{G}_h^N)$, respectively, which are assumed to be known and continuous. The goals of the leader and followers are to maximize their value functions:

$$V^L(\mu, (\pi^i)_{i\in\mathcal{N}}) = \mathbb{E}\Big[\sum\nolimits_{h=0}^{H} r^L(x_h, u_h, \mathbb{G}_h^N)\Big| x_0 \sim \rho_0^L, \\ u_h \sim \mu, x_{h+1} \sim P^L\Big],$$

$$V^i(\mu, (\pi^i)_{i\in\mathcal{N}}) = \mathbb{E}\Big[\sum\nolimits_{h=0}^{H} r^i(u_h, s_h^i, a_h^i, \mathbb{G}_h^N)\Big| u_h \sim \mu, \\ s_0^i \sim \rho_0, a_h^i \sim \pi^i, s_{h+1}^i \sim P\Big],$$

where ρ_0^L and ρ_0 are respectively the initial state distributions for the leader and followers.

Given that all the followers are homogeneous (i.e., share the same state and action spaces as well as the transition and reward functions), when $N \to \infty$, mean-field game (MFG) (Guo et al. 2019; Huang et al. 2006) can be used to model the interactions among them. The MFG consists of the same elements as the finite followers case. However, instead of modeling N followers separately (each follower i uses a distinct policy π^i), it models a representative follower and collapses all other followers into their statistical state distribution, which is called the mean-field and denoted by $m_h \in \Delta(\mathcal{S})$. Formally, $\mathbb{G}_h^N$ converges to m_h as $N \to \infty$ due to the strong law of large numbers: $\forall k \in \mathcal{S}$,

$$m_h(k) = \lim_{N\to\infty} \mathbb{G}_h^N(k) = \lim_{N\to\infty} \frac{1}{N} \sum\nolimits_{i\in\mathcal{N}} \mathbf{1}_k(s_h^i). \tag{1}$$

Let $m = (m_h)_{h\in\mathcal{H}} \in \mathcal{M}$, where $\mathcal{M}$ is the space of all mean-fields. With the mean-field, we will focus on a representative follower and omit the follower index i henceforth. Then the rewards for the leader and followers are $r^L(x_h, u_h, m_h)$ and $r(u_h, s_h, a_h, m_h)$, which depend on m_h. We adopt the setting in (Lasry and Lions 2007; Perrin et al. 2020; Perolat et al. 2022) where r satisfies the following conditions. (1) Separability: $\forall h \in \mathcal{H}$, $u \in \mathcal{U}$, $r(u, s, a, m_h) = \tilde{r}(s, a) + \bar{r}(u, s, m_h)$. That is, the reward for a follower consists of two parts: the first part is only related to her own state and action and the second part is determined by the leader's action and her state as well as the mean-field of the followers. It is a natural depiction of various systems, e.g., in EDRP scenario, $\tilde{r}(s, a)$ is the cost of taking action a in state s and $\bar{r}(u, s, m_h)$ is the order price. (2) Monotonicity: $\forall m_h^1, m_h^2 \in \Delta(\mathcal{S})$, $\sum_{s\in\mathcal{S}}(m_h^1(s) - m_h^2(s))(\bar{r}(u, s, m_h^1) - \bar{r}(u, s, m_h^2)) < 0$. That is, given leader's action $u \sim \mu$, the reward of a follower decreases with the increase of the number of other followers presenting in the same state. It shows followers' aversion to crowded areas (e.g., the zones with more drivers in EDRP scenario), which is a common phenomenon in practice. We will show that these conditions are satisfied in our experimental environments (see the Appendix for discussion).

Given π and the initial mean-field $m_0 \in \Delta(\mathcal{S})$, the mean-field induced by this policy is defined by the Fokker-Planck (FP) equation: $\forall k \in \mathcal{S}$,

$$\begin{aligned} m_{h+1}^{\pi}(k) &= \Phi(\pi, m_h^{\pi})(k) \\ &= \sum\nolimits_{l\in\mathcal{S}} \sum\nolimits_{a_h\in\mathcal{A}} m_h^{\pi}(l)\pi(a_h|l)P(k|l, a_h). \end{aligned} \tag{2}$$

Given (μ, π, m) and the initial states of the leader $x_0 \sim \rho^L$ and the representative follower $s_0 \sim m_0$, the value functions for the leader and the representative follower are:

$$\begin{aligned} V^L(\mu, \pi, m) &= \mathbb{E}\Big[\sum\nolimits_{h=0}^{H} r^L(x_h, u_h, m_h)\Big| x_0 \sim \rho^L, \\ &\quad u_h \sim \mu, m_{h+1}^{\pi}(k) = \Phi(\pi, m_h^{\pi})(k), \forall k \in \mathcal{S}\Big], \\ V(\mu, \pi, m) &= \mathbb{E}\Big[\sum\nolimits_{h=0}^{H} r(u_h, s_h, a_h, m_h)\Big| u_h \sim \mu, \\ &\quad s_0 \sim m_0, a_h \sim \pi, m_{h+1}^{\pi}(k) = \Phi(\pi, m_h^{\pi})(k), \forall k \in \mathcal{S}\Big]. \end{aligned}$$

Given $\mu \in \Upsilon$, we call π_μ^* a Nash Equilibrium of the followers induced by μ if $V(\mu, \pi_\mu^*, m_\mu^*) \geq V(\mu, \pi, m_\mu^*)$, $\forall \pi \in \Pi$, i.e., no follower has an incentive to deviate (Muller et al. 2022). For notation simplicity, we use $m_\mu^* = m^{\pi_\mu^*}$, then the mean-field of the followers following π_μ^*. Then, the Stackelberg Nash Equilibrium (SNE) is defined as follows.

Definition 1. $(\mu^*, \pi_{\mu^*}^*)$ *is an SNE if it satisfies: i) leader's optimality:* $\mu^* \in \arg\max_{\mu\in\Upsilon} V^L(\mu, \pi_\mu^*, m_\mu^*)$*, where* π_μ^* *is the NE induced by* μ *and* m_μ^* *is the mean-field of the followers following* π_μ^**; ii) followers' NE induced by* μ^**:* $V(\mu^*, \pi_{\mu^*}^*, m_{\mu^*}^*) \geq V(\mu^*, \pi, m_{\mu^*}^*)$*,* $\forall \pi \in \Pi$*, with* $m_{\mu^*}^*$ *being the mean-field of the followers following* $\pi_{\mu^*}^*$*.*

In other words, in SNE, the leader chooses an optimal policy, given that the followers will reach the corresponding NE induced by her optimal policy.

A conclusion is that *for any given leader's policy* $\mu \in \Upsilon$*, there exists a unique NE of the followers* π_μ^* *(i.e., no tie-breaking rule is involved).* This follows the convergence of a well-defined fictitious play process and the uniqueness is derived by the monotonicity of r (see the Appendix).

With the above result, the leader can get rid of the equilibrium selection problem, which is one of the challenges in game theory (Samuelson 1997). Let d be some distance metric defined on $\Upsilon \times \Pi$, $(\mu^1, \pi_{\mu^1}^*)$ and $(\mu^2, \pi_{\mu^2}^*)$ be two different policy profiles with $m_{\mu^1}^*$ and $m_{\mu^2}^*$ being the corresponding mean-fields. Then, if V^L is concave w.r.t. μ, we can get that: $V^L(\mu^1, \pi_{\mu^1}^*, m_{\mu^1}^*) \geq V^L(\mu^2, \pi_{\mu^2}^*, m_{\mu^2}^*)$ implies that $d((\mu^1, \pi_{\mu^1}^*), (\mu^*, \pi_{\mu^*}^*)) \leq d((\mu^2, \pi_{\mu^2}^*), (\mu^*, \pi_{\mu^*}^*))$. In other words, for any policy profile (μ, π_μ^*), the higher (or equal) leader's value means the closer (or equal) distance of this profile to the SNE under the concavity of V^L. This motivates our method for evaluating any leader's policy μ: we fix μ and then re-calculate the followers' policy such that they achieve the unique NE π_μ^*, after which we report the leader's performance. Such an evaluation method is similar to that in Stackelberg security games where the defender maximizes her utility given that the attacker best responds to her strategies (Kar et al. 2017). As we aim to find a good leader's policy, such evaluation is reasonable in our work.

To facilitate the understanding of this work, we summarize some key notations in Table 1.

Notation	Description
$x_h/\mathcal{X}$, $u_h/\mathcal{U}$	Leader's state/state space, action/action space
$s_h^i/\mathcal{S}$, $a_h^i/\mathcal{A}$	Follower i's state/state space, action/action space
$\mathbb{G}_h^N$	Followers' empirical state distribution
P^L/P	Leader's/Followers' transition function
μ/π^i, Υ/Π	Leader's/Follower i's policy, policy space
r^L/r^i	Leader's/Follower i's reward function
V^L/V^i	Leader's/Follower i's value function
$m_h/\mathcal{M}$	Followers' mean-field/mean-field space

Table 1: Summary of notations. The notations for the representative follower are similar and hence, omitted here.

3.2 Example: E-hailing Driver Re-Positioning

We use the e-hailing driver re-positioning (EDRP) to intuitively illustrate the game model presented in the previous section. Consider a city partitioned into different districts.

Each district is denoted as a node, and the adjacent relationship between nodes is represented by edges. Then the city is abstracted as a graph $\mathcal{G} = (V, E)$ where V and E are the sets of nodes and edges, respectively. The leader is the e-hailing platform such as Uber and Lyft, and the followers are the taxi drivers who compete for orders through the platform.

The state space of the followers is the set of nodes $\mathcal{S} = \{1, 2, \cdots, |V|\}$ where $|V|$ is the number of nodes. The set of available actions for a representative follower in state $s_h = v \in \mathcal{S}$ is denoted by $\mathcal{A}^v \subseteq V$. At time $h \in \mathcal{H}$, the leader observes the followers' state distribution $\mathbb{G}_h^N$ and maximizes the order response rate (ORR) of the city by placing service charge in the nodes which are oversupplied by the followers. The leader first takes an action u_h and then, the representative follower selects the next node $a_h = w \in \mathcal{A}^v$ that she will travel to based on her current state $s_h = v$. Then, the follower changes to a new state. The leader and follower receive $r^L(x_h, u_h, \mathbb{G}_h^N)$ and $r(u_h, s_h, a_h, \mathbb{G}_h^N)$, respectively.

In real-world scenarios, the number of followers could be extremely large (e.g., ∼13,000 taxis in Manhattan) and they share the same state/action space and transition/reward function. Therefore, SMFG can be used to model this scenario in which the interaction between the leader and followers is modeled as a Stackelberg game and the interaction between the followers is captured by an MFG.

4 Approaches

In this section, we first propose an RL framework which we call the Stackelberg mean-field update (SMFU) to learn the leader's policy. Then, as the SMFU is a model-free algorithm which is data-inefficient, we propose a novel learning framework, the transition-informed reinforcement learning (TIRL), that improves the data efficiency and in turn accelerates the learning process. Finally, we propose to use regularization techniques to further improve the stability.

4.1 Stackelberg Mean-Field Update

Classical methods for solving the SMFG typically require exact dynamics parameters (e.g., the exact transition rate matrix) which are unavailable in many real-world scenarios and hence, are infeasible in practice. On the contrary, as RL can learn the equilibrium without any prior of environments, it has been widely used to solve complex multi-agent problems. However, none of the existing works have tried to solve the SMFG by using RL algorithms. In this section, we propose the first RL framework for solving the SMFG. Suppose that the leader's policy is parametrized by $\omega \in \Omega$ and the followers' is parametrized by $\theta \in \Theta$. Moreover, the leader's and followers' initial states come from the initial distributions d_0^L and d_0, respectively. Then the leader's and followers' expected performance is given by[1]:

$$J^L(\omega, \theta, m^\theta) = \mathbb{E}_{x_0 \sim d_0^L} V^L(\omega, \theta, m^\theta),$$

$$J(\omega, \theta, m^\theta) = \mathbb{E}_{s_0 \sim d_0} V(\omega, \theta, m^\theta).$$

Let $\nabla_\omega J^L(\omega, \theta, m^\theta)$ and $\nabla_\theta J(\omega, \theta, m^\theta)$ be the unbiased estimators of $\partial J^L(\omega, \theta, m^\theta)/\partial\omega$ and $\partial J(\omega, \theta, m^\theta)/\partial\theta$, respectively. Then, the learning rule for computing leader's policy, which we refer to as the Stackelberg mean-field update (SMFU), is given as follows:

$$\omega_{h+1} = \omega_h + \alpha_h \nabla_\omega J^L(\omega_h, \theta_h, m^{\theta_h}), \tag{3}$$

$$\theta_{h+1} = \theta_h + \beta_h \nabla_\theta J(\omega_h, \theta_h, m^{\theta_h}), \tag{4}$$

$$m_h^{\theta_h} \approx \mathbb{G}_h^{N, \theta_h}, \tag{5}$$

where α_h and β_h are the learning rates. In practice, the gradients of ω and θ are estimated by sampling experiences from interactions with the environment (simulator). Take the leader as the example, an unbiased estimator is:

$$\nabla_\omega J^L(\omega, \theta, m^\theta) = \mathbb{E}_{\mu_\omega}\Big[\nabla_\omega \log \mu_\omega(u_h|x_h) \times \sum\nolimits_{z=0}^{H} r^L(x_{h+z}, u_{h+z}, m_{h+z}^\theta)\Big].$$

As for m_h^θ, according to Eq.(1), we can approximate it by $\mathbb{G}_h^{N,\theta}$, the followers' empirical state distribution at $h \in \mathcal{H}$.

4.2 Transition-Informed Reinforcement Learning

Since the SMFU is a model-free learning algorithm, it suffers from the issue of data inefficiency. On one hand, model-free learning typically requires an excessive number of interactions of the players with the environment to generate a large amount of data before achieving satisfied performance. Though the availability of a large number of followers that explore the environment can generate the desired data, the simulation of the behaviors of a large population of followers is computationally resource-intensive and could be time-consuming. On the other hand, the experience tuples generated by interacting with the environment are ignored after each update of ω and θ. However, in fact, extra information can be extracted from the data, which can be used for facilitating the future updates of the policies. This motivates our novel learning framework, the Transition-Informed Reinforcement Learning (TIRL), which is shown in Figure 1.

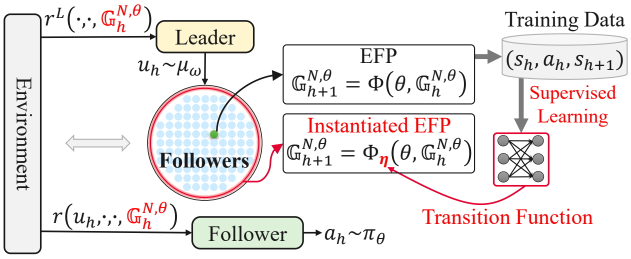
$r^L(\cdot,\cdot, \mathbb{G}_h^{N,\theta})$
Leader
$u_h \sim \mu_\omega$
Environment
Followers
EFP
$\mathbb{G}_{h+1}^{N,\theta} = \Phi(\theta, \mathbb{G}_h^{N,\theta})$
Instantiated EFP
$\mathbb{G}_{h+1}^{N,\theta} = \Phi_\eta(\theta, \mathbb{G}_h^{N,\theta})$
Training Data
(s_h, a_h, s_{h+1})
Supervised Learning
Transition Function
$r(u_h,\cdot,\cdot, \mathbb{G}_h^{N,\theta})$
Follower
$a_h \sim \pi_\theta$

Figure 1: Overview of TIRL.

Roughly speaking, there are two stages in TIRL: i) in addition to training the leader's and followers' policies, we use the experience tuples generated by interacting with the environment to learn the followers' transition function, and ii) train the leader's and followers' policies based on the empirical Fokker-Planck (EFP) equation (defined below) instantiated with the learned transition function. More details on the learning process can be found in the Appendix. To formally describe the TIRL, we introduce the EFP equation under the

[1] For simplicity, we use ω and θ to represent μ_ω and π_θ.

parameterized policy π_θ: $\forall k \in \mathcal{S}$,

$$\mathbb{G}_{h+1}^{N,\theta}(k) = \sum\nolimits_{l\in\mathcal{S}} \sum\nolimits_{a_h\in\mathcal{A}} \mathbb{G}_h^{N,\theta}(l)\pi_\theta(a_h|l)P(k|l,a_h). \tag{6}$$

Intuitively, Eq.(6) describes the evolution of the followers' empirical state distribution given a large number of followers. Notice that rigorously proving the equivalence between Eq.(6) and Eq.(2) under the parameterized policy π_θ is more involved and outside the scope of this work (please refer to the Appendix for more discussion).

Let P_η be the neural network (NN) parameterized by η, which acts as the approximator of the followers' ground-truth transition function P. P_η takes s_h and a_h as input and outputs the probability distribution over the state space. Let K be the number of episodes of the whole learning process and $K_1 < K$ be the number of episodes for learning the followers' transition function. During the first K_1 episodes, we collect data of form (s_h, a_h, s_{h+1}) to the training set $\mathcal{D}$ by interacting with the environment and train P_η on $\mathcal{D}$ via negative log prediction probability loss (supervised learning):

$$\text{loss}(\eta) = -\mathbb{E}_{\mathcal{D}}\left[\log P_\eta(s_{h+1}|s_h,a_h)\right]. \tag{7}$$

Then, for last $K - K_1$ episodes, we generate data by leveraging Eq.(6) where P is replaced with P_η. Overall, we have the following update rules: $\forall k \in \mathcal{S}$,

$$\omega_{h+1} = \omega_h + \alpha_h \nabla_\omega J^L(\omega_h, \theta_h, \mathbb{G}^{N,\theta_h}), \tag{8}$$

$$\theta_{h+1} = \theta_h + \beta_h \nabla_\theta J(\omega_h, \theta_h, \mathbb{G}^{N,\theta_h}), \tag{9}$$

$$\mathbb{G}_{h+1}^{N,\theta}(k) = \begin{cases} \Phi(\theta, \mathbb{G}_h^{N,\theta})(k), & e \le K_1, \\ \Phi_\eta(\theta, \mathbb{G}_h^{N,\theta})(k), & \text{otherwise}, \end{cases} \tag{10}$$

where Φ_η denotes the operation with P_η and $1 \le e \le K$ denotes the current episode. Note that when $e \le K_1$, since P is unknown (to algorithm), Φ is implemented through interactions with the environment. That is, $\mathbb{G}_{h+1}^{N,\theta}$ is obtained by taking statistics on s_{h+1}^i, $i \in \mathcal{N}$, the followers' new states resulted from interactions with the environment by using π_θ. During the last $K - K_1$ episodes, in contrast to the SMFU, TIRL estimates the gradients of ω and θ by directly considering the transition happened in every pair of states $(l, k) \in \mathcal{S} \times \mathcal{S}$. By the instantiated EFP equation, we can directly compute $\mathbb{G}_{h+1}^{N,\theta}$, based on which we obtain the reward for each state transition. Specifically, when computing the new state distribution of followers in state k, we follow Eq.(10) to first multiply the current state distribution with action probabilities and $P_\eta(k|\cdot,\cdot)$ which can be easily obtained in a batch manner, and then summarize. Note that the current state distribution and action probabilities are expanded to match the dimension of $P_\eta(k|\cdot,\cdot)$.

Obviously, the computational complexity of TIRL is proportional to the size of the followers' state space $|\mathcal{S}|$. Since the number of states could be much smaller than that of the followers ($|\mathcal{S}| \ll N$), e.g., the number of districts of the city versus the number of vehicles, TIRL can therefore scale up to much larger settings.

4.3 Regularized TIRL

The complex interaction between the leader and followers makes the learning performance of the TIRL sensitive to the initialization of the leader's policy. We empirically demonstrate this phenomenon in the EDRP scenario as shown in Figure 2 (colored lines correspond to the evaluation performance of leader's policies learned by using TIRL under different random seeds and the black line for SMFU-w/o-L is plotted for comparison, details on experimental settings can be found in Section 5). From the results, we can see that different initializations of leader's policy can result in dramatically different learning performance. To alleviate this issue, we develop a regularized TIRL (Reg-TIRL, for short).

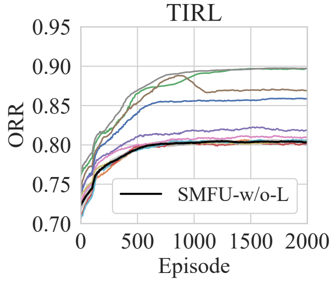

Figure 2: Test results of 10 seeds in EDRP with $N = 1,000$.

First, to encourage the exploration of leader's policy and prevent it from overfitting to some of the actions, we augment the leader's value function with an entropy term calculated as $I(\mu_\omega(\cdot|x_h)) = -\lambda \mathbb{E}_{u_h \sim \mu_\omega(u_h|x_h)} \log \mu_\omega(u_h|x_h)$, where λ determines the relative importance of the entropy term against the value function. Though entropy regularization can improve stability, we find that it is still insufficient to stabilize the leader's policy training. We address this issue by further constraining the complexity of the leader's policy network. Specifically, we augment the leader's objective with the L_1-norm $\delta\|\omega\|_1$ where δ is a hyperparameter. After each update, we apply weight clipping to constraint the weights of the leader's policy to $\omega = \text{clip}(\omega, -c, c)$. Note that the method can be also applied to SMFU (Reg-SMFU).

5 Experimental Results

We evaluate our approach on two scenarios[2]: the e-hailing driver re-positioning (EDRP) and multiple-type food gathering (MTFG). All experiments are run on a 64-bit workstation with 125 GB RAM, 20 Intel i9-9820X CPU @3.30GHz processors, and 4 NVIDIA RTX2080 Ti GPUs.

5.1 Setup

First, we present the experimental setup including the evaluation metrics, baseline methods, and game environments.

Evaluation Metrics. For training, we focus on the runtime of the training process. For evaluation, we concentrate on the leader's performance; thus, we fixed the leader's policy and then re-train the followers' policy for a fixed number of

[2] Code is available at https://github.com/IpadLi/SMFG.

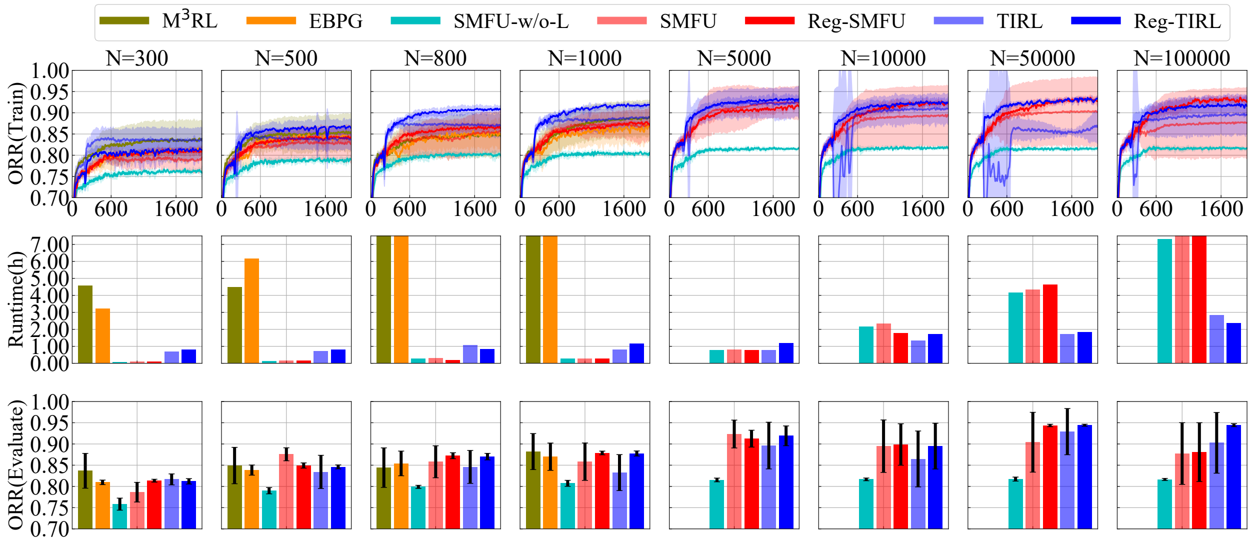

Figure 3: Results for EDRP scenario. (Top) Training curves of different methods. x-axis is episode. (Medium) Average runtimes of different methods, 'h' stands for hour. (Bottom) Evaluation performance of the leader's policies of different methods.

N	M³RL	EBPG	SMFU-w/o-L	SMFU	Reg-SMFU	TIRL	Reg-TIRL
300	0.84±0.03	0.81±0.01	0.76±0.00	0.79±0.02	**0.81±0.00**	0.82±0.02	**0.81±0.01**
500	0.85±0.04	0.84±0.01	0.79±0.01	0.88±0.02	**0.85±0.00**	0.83±0.04	**0.85±0.01**
800	0.84±0.05	0.85±0.03	0.80±0.01	0.86±0.04	**0.87±0.01**	0.85±0.04	**0.87±0.01**
1,000	0.88±0.04	0.87±0.03	0.81±0.01	0.86±0.04	**0.88±0.01**	0.83±0.04	**0.88±0.01**
5,000	N/A	N/A	0.82±0.00	0.92±0.03	**0.91±0.02**	0.90±0.06	**0.92±0.02**
10,000	N/A	N/A	0.82±0.01	0.89±0.06	**0.90±0.06**	0.86±0.06	**0.89±0.05**
50,000	N/A	N/A	0.82±0.01	0.90±0.07	**0.94±0.01**	0.93±0.06	**0.94±0.00**
100,000	N/A	N/A	0.82±0.00	0.88±0.07	**0.88±0.07**	0.90±0.07	**0.94±0.00**

Table 2: Quantitative values of the performance of leader's policies in EDRP scenario. ± corresponds to the standard deviation.

episodes, after which we report the leader's average performance over the last 10 episodes for fair comparison.

Baselines. We consider the following methods: M³RL (Shu and Tian 2019), EBPG (Yu et al. 2020), SMFU-w/o-L (i.e., without leader's incentive), SMFU, Reg-SMFU, TIRL, and Reg-TIRL. As M³RL and EBPG require much longer training/evaluation time, we only run them for $N \leq 1,000$, while for the other methods, we run them up to the setting with $N = 100,000$. All the settings are run using 5 seeds. More details can be found in the Appendix.

Environments. (1) The EDRP environment is adapted from (Lin et al. 2018). In this scenario, the leader aims to improve the order response rate (ORR) of the whole city, while the followers maximize their own returns. Similar scenarios can be found in (Varakantham et al. 2012; Nguyen, Kumar, and Lau 2017, 2018). Moreover, similar to (Alonso-Mora et al. 2017), we extract order information from a public dataset of taxi trips in Manhattan, which contains for each day the time and location of all the pickups and drop-offs executed by each of ∼13,000 active taxis. Such information can be seen as an approximation of the real-world demand-supply relationship in Manhattan. (2) The MTFG environment is adapted from (Long et al. 2019). In this scenario, two types of foods with different values are distributed over the map. The followers gain benefits by gathering as many foods as possible and the leader manages to improve the collection ratio (CR) of the whole system by offering the followers incentives. See the Appendix for more details.

5.2 Results

EDRP. In Figure 3, we show the training curves (top), average runtimes (medium), and evaluation performance (bottom) of the leader's policies under different settings. **Top**: i) In small-sized settings, TIRL and Reg-TIRL perform better than the other methods. Intuitively, TIRL and Reg-TIRL operate on the mean-field level (i.e., directly compute the followers' state distribution), which can more accurately approximate the order distribution so that they achieve better ORR. As the number of followers increases, the training performance of the SMFU-type methods (SMFU-w/o-L, SMFU, and Reg-SMFU) increases due to the fact that the empirical state distribution of the followers converges to the mean-field. ii) Using regularization techniques can significantly improve the training stability, e.g., in the set-

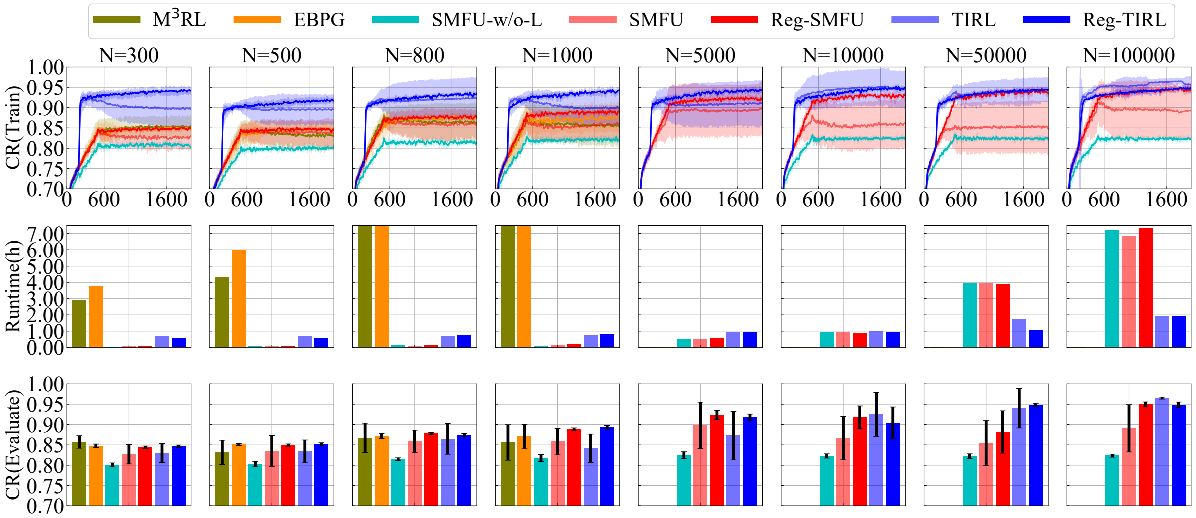

Figure 4: Results for MTFG scenario. (Top) Training curves of different methods. x-axis is episode. (Medium) Average runtimes of different methods, 'h' stands for hour. (Bottom) Evaluation performance of the leader's policies of different methods.

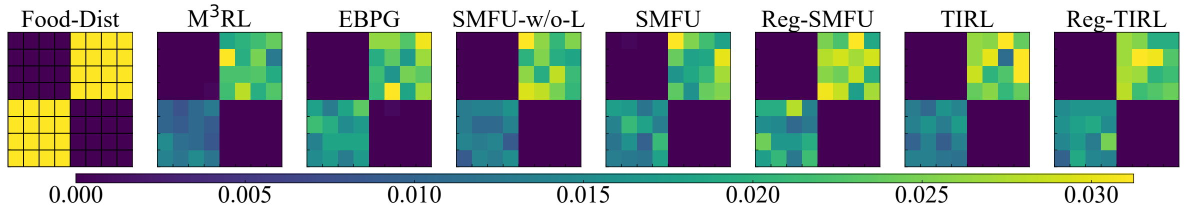

Figure 5: First column is the distribution of foods (the two types of foods are equally deployed on the bottom-left and top-right zones, respectively). The other columns are the followers' distributions of different methods. Here, $N = 1,000$.

ting with $N = 50,000$, Reg-TIRL performs better than TIRL. Similar results can be found between Reg-SMFU and SMFU. **Medium**: The runtimes of M^{3}RL and EBPG are much longer than all the other methods. As the number of followers increases, the runtimes of the SMFU-type methods increase quickly, while TIRL and Reg-TIRL need shorter runtime, showing that they can scale up to larger settings. **Bottom**: i) TIRL and Reg-TIRL achieve similar or better ORR compared to SMFU and Reg-SMFU, respectively. ii) Using regularization significantly improves learning stability, see the comparison of Reg-SMFU *vs.* SMFU and Reg-TIRL *vs.* TIRL. In conclusion, TIRL/Reg-TIRL can efficiently (shorter runtime) and effectively (better leader's performance) solve large-scale SMFGs (scale up to 100,000 followers). Quantitative values can be found in Table 2.

In addition, we also perform the ablation study on a small-sized EDRP setting with $N = 1,000$ to show that Reg-TIRL achieves the best or similar performance compared to some other common regularizers: L_2-norm, batch normalization, and dropout. See the Appendix for more details.

MTFG. In Figure 4, we show the training curves (top), average runtimes (medium), and the evaluation performance (bottom) of the leader's policies. From the results, we observe that TIRL/Reg-TIRL can efficiently (shorter runtime) and effectively (better leader's performance) solve large-scale SMFGs (scale up to 100,000 followers). Quantitative values can be found in the Appendix. Moreover, in Figure 5, we show the distributions of the followers over the grid map, which demonstrates the effectiveness of the leader's policy on incentivizing a large population of followers. Here only the setting with $N = 1,000$ is shown and more results can be found in the Appendix.

6 Conclusions

In this work, we employ RL to address large-scale SMFGs. The main contributions are threefold: i) we propose an RL framework, the SMFU, to learn the leader's policy without priors of the environments, ii) we propose a transition-informed RL framework to improve the data efficiency and in turn, the scalability, and iii) we propose to use regularization techniques to improve the training stability. Extensive experiments on real-world domains demonstrate the effectiveness of our approach. Some limitations of our approach will be further investigated in future works: i) we will consider multiple leaders and multiple followers, which is the case in the labor market, and ii) we will consider the setting with enormous multiple-type, even heterogeneous, followers (Zheng et al. 2022), such as different types of vehicles (e.g., cars, buses, and trucks) in the EDRP scenario.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Prepositioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Alonso-Mora, J.; Samaranayake, S.; Wallar, A.; Frazzoli, E.; and Rus, D. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *PNAS*, 114(3): 462–467.

Aumann, R. J.; and Shapley, L. S. 2015. *Values of Non-Atomic Games*. Princeton University Press.

Aurell, A.; Carmona, R.; Dayanikli, G.; and Lauriere, M. 2022. Optimal incentives to mitigate epidemics: A Stackelberg mean field game approach. *SIAM Journal on Control and Optimization*, 60(2): S294–S322.

Bensoussan, A.; Chau, M.; Lai, Y.; and Yam, S. C. P. 2017. Linear-quadratic mean field Stackelberg games with state and control delays. *SIAM Journal on Control and Optimization*, 55(4): 2748–2781.

Cai, Q.; Filos-Ratsikas, A.; Tang, P.; and Zhang, Y. 2018. Reinforcement mechanism design for e-commerce. In *WWW*, 1339–1348.

Fu, G.; and Horst, U. 2020. Mean-field leader-follower games with terminal state constraint. *SIAM Journal on Control and Optimization*, 58(4): 2078–2113.

Guo, X.; Hu, A.; Xu, R.; and Zhang, J. 2019. Learning mean-field games. In *NeurIPS*, 4967–4977.

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

Huang, M.; and Yang, X. 2020. Mean field Stackelberg games: State feedback equilibrium. *IFAC-PapersOnLine*, 53(2): 2237–2242.

Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2020. When to trust your model: Model-based policy optimization. In *NeurIPS*, 12519–12530.

Kamra, N.; Zhu, H.; Trivedi, D. K.; Zhang, M.; and Liu, Y. 2020. Multi-agent trajectory prediction with fuzzy query attention. In *NeurIPS*, 22530–22541.

Kar, D.; Nguyen, T. H.; Fang, F.; Brown, M.; Sinha, A.; Tambe, M.; and Jiang, A. X. 2017. Trends and applications in Stackelberg security games. *Handbook of Dynamic Game Theory*, 1–47.

Krupnik, O.; Mordatch, I.; and Tamar, A. 2020. Multi-agent reinforcement learning with multi-step generative models. In *CoRL*, 776–790.

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

Lin, K.; Zhao, R.; Xu, Z.; and Zhou, J. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *KDD*, 1774–1783.

Long, Q.; Zhou, Z.; Gupta, A.; Fang, F.; Wu, Y.; and Wang, X. 2019. Evolutionary population curriculum for scaling multi-agent reinforcement learning. In *ICLR*.

Muller, P.; Rowland, M.; Elie, R.; Piliouras, G.; Perolat, J.; Lauriere, M.; Marinier, R.; Pietquin, O.; and Tuyls, K. 2022. Learning equilibria in mean-field games: Introducing mean-field PSRO. In *AAMAS*, 926–934.

Nguyen, D. T.; Kumar, A.; and Lau, H. C. 2017. Policy gradient with value function approximation for collective multiagent planning. In *NeurIPS*, 4319–4329.

Nguyen, D. T.; Kumar, A.; and Lau, H. C. 2018. Credit assignment for collective multiagent RL with global rewards. In *NeurIPS*, 8102–8113.

Perolat, J.; Perrin, S.; Elie, R.; Laurière, M.; Piliouras, G.; Geist, M.; Tuyls, K.; and Pietquin, O. 2022. Scaling up mean field games with online mirror descent. In *AAMAS*, 1028–1037.

Perrin, S.; Pérolat, J.; Laurière, M.; Geist, M.; Elie, R.; and Pietquin, O. 2020. Fictitious play for mean field games: Continuous time analysis and applications. In *NeurIPS*, 13199–13213.

Qiu, W.; Chen, H.; and An, B. 2019. Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks. In *IJCAI*, 4568–4574.

Samuelson, L. 1997. *Evolutionary Games and Equilibrium Selection*, volume 1. MIT press.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Shou, Z.; and Di, X. 2020. Reward design for driver repositioning using multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 119: 102738.

Shu, T.; and Tian, Y. 2019. M^{3}RL: Mind-aware multi-agent management reinforcement learning. In *ICLR*.

Subramanian, J.; and Mahajan, A. 2019. Reinforcement learning in stationary mean-field games. In *AAMAS*, 251–259.

Varakantham, P.; Cheng, S.-F.; Gordon, G.; and Ahmed, A. 2012. Decision support for agent populations in uncertain and congested environments. In *AAAI*, 1471–1477.

Yu, R.; Wang, X.; Zhang, Y.; Wang, R.; An, B.; Shi, Z.; and Lai, H. 2020. Learning expensive coordination: An event-based deep RL approach. In *ICLR*.

Zhang, W.; Wang, X.; Shen, J.; and Zhou, M. 2021. Model-based multi-agent policy optimization with adaptive opponent-wise rollouts. In *IJCAI*, 3384–3391.

Zheng, S.; Trott, A.; Srinivasa, S.; Parkes, D. C.; and Socher, R. 2022. The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18): eabk2607.