Modelling Competitive Behaviors in Autonomous Driving Under Generative World Model

Guanren Qiao[1], Guorui Quan[2], Rongxiao Qu[1], and Guiliang Liu[1]*

[1] The Chinese University of Hong Kong, Shenzhen
[2] The University of Manchester
*Correspondence to: Guiliang Liu, liuguiliang@cuhk.edu.cn

Abstract. Modeling the trajectories of intelligent vehicles is an essential component of a traffic-simulating system. However, such trajectory predictors are typically trained to imitate the movements of human drivers. The imitation models often fall short of capturing safety-critical events residing in the long-tail end of the data distribution, especially under complex environments involving multiple drivers. In this paper, we propose a game-theoretic perspective to resolve this challenge by modeling the competitive interactions of vehicles in a general-sum Markov game and characterizing these safety-critical events with the correlated equilibrium. To achieve this goal, we pretrain a generative world model to predict the environmental dynamics of self-driving scenarios. Based on this world model, we probe the action predictor for identifying the Coarse Correlated Equilibrium (CCE) by incorporating both optimistic Bellman update and magnetic mirror descent into the objective function of the Multi-Agent Reinforcement Learning (MARL) algorithm. We conduct extensive experiments to demonstrate our algorithm outperforms other baselines in terms of efficiently closing the CCE-gap and generating meaningful trajectories under competitive autonomous driving environments. The code is available at: https://github.com/qiaoguanren/MARL-CCE.

Keywords: Multi-Agent Reinforcement Learning · Generative World Model · Coarse Correlated Equilibrium · Traffic Simulation

1 Introduction

Realistic traffic simulation plays an indispensable role in the development of self-driving software [26]. It provides a safe and scalable environment for refinement and testing before deployment in real-world scenarios. Previous simulation systems typically predict vehicles' trajectories by imitating realistic driver behaviors from an offline dataset [3, 9, 22, 34, 37, 38, 41, 42]. While this approach can precisely replicate common occurrences, it often falls short in capturing Out-of-Distribution (OoD) and long-tail events. On the other hand, modern self-driving agents are capable of managing routine traffic scenarios [45]; however, they often fail to handle safety-critical events in the long tail end of data distributions [42]. This dichotomy underscores a significant challenge: traffic simulators are *proficient at*

modeling scenarios that have already been addressed by self-driving algorithms but *struggle to represent the critical and as-yet-unsolved situations accurately* that are vital for advancing the robustness of self-driving.

To more accurately simulate the long-tail or OoD events, prior works often refine their generation process of these events with safety mapping network [42] or traffic compliance constraint [48]. These refinement strategies often focus exclusively on a single target agent, neglecting the *dynamic interactions* where surrounding vehicles respond to the target's movements. In real-world conditions, however, safety-critical events (e.g., emergency braking, sudden lane changes, and merging) often involve multiple vehicles, and human drivers can regularly adjust their behavior in reaction to the actions of the self-driving car. This oversight results in a persistent simulation-to-reality (sim-to-real) gap.

In this work, we develop a game-theoretic approach to generate critical events by viewing traffic simulation as a general-sum Markov game involving multiple agents. To better represent the realistic driving scenarios, our Makrov game is based on a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) where each car can observe and interact with the vehicles in a confined region, and the goal of each agent is to reach their destination within a minimum time. The objective of our Markov game is to learn the Coarse Correlated Equilibrium (CCE) in the competitive game while maintaining fidelity to realistic human behaviors. These CCEs can characterize the competitive interactions of vehicles in an interdependent manner so that the simulated vehicles can react to the ego car's movements, which better represents critical events in real traffic.

To efficiently learn CCE, we construct a world model from an offline dataset to represent the environmental dynamics (e.g., action predictor, transition, and observation model) in Dec-POMDP. This world model is implemented by multi-modal cross-attention layers to represent the agents' interaction with the scene map, historical events, and surrounding vehicles. Within this world model, Multi-Agent Reinforcement Learning (MARL) methods [24] can be incorporated to capture the CCE. To achieve this goal, we integrate both 1) the optimistic Bellman update [2] and 2) the magnetic mirror decent [35] into our learning objective. Their theoretical property can guarantee the optimality and fidelity of MARL agents in competitive games. Derived from these objectives, we design a Coarse Correlated Equilibrium Multi-agent Soft Actor-Critic algorithm (CCE-MASAC) by following the Centralized Training with Decentralized Execution (CTDE) framework [27] where the critic and actor functions for each agent are separately parameterized and iteratively updated.

Our experimental results demonstrate the superior performance of our methodology in high-dimensional traffic simulation domains. The CCE-MASAC algorithm consistently surpasses competing baselines by effectively minimizing the CCE-gap. To demonstrate the capability of the generative world model in enabling optimal control, we delve into a detailed examination of both single-agent and multi-agent control performances within this world model. Our analysis reveals a distinct gap in controlling performance that arises from the competitive interactions between agents. To illustrate how well the CCE captures safe-critical events,

we visualize the simulated traffic in several complex and competitive scenarios. These simulations provide an intuitive and clear illustration of our approach's effectiveness in modeling competitive behaviors.

2 Related Works

The works that are most related to our approach are introduced as follows:
Learning in General-Sum Games. In the general-sum game, agents often have competing incentives [43]. In general, general-sum game solvers can be divided into 1) value-based approaches and 2) policy-based approaches. The value-based solvers commonly extend classic RL algorithms into the multi-agent setting. These methods include Nash Q-learning [14], Optimistic Nash Q/V-learning [2], CCE/CE V-learning algorithms [17, 36], and Online Mirror Descent (OMD) V-learning [28]. The policy-based methods solve general-sum games from a policy gradient perspective. [47] explored direct parameterization of policy for capturing Nash Equilibrium (NE) in general-sum stochastic games. [20] introduced the concept of Markov potential games for general-sum stochastic games, and employed REINFORCE gradient estimators for learning a NE. Previous methods commonly study NE under toy games, whereas the CCE solvers in practical applications and complex game contexts are less explored.
Realistic Traffic Simulation. Traffic simulation techniques can be broadly categorized into rule-based and learning-based approaches. Rule-based strategies [8, 18] rely on user interfaces for defining vehicle routes, with motion regulated by analytical models such as the Intelligent Driver Model [4]. This rigidity limits their ability to characterize real-world driving behaviors accurately. To improve realism, learning-based approaches [21, 39] predict or simulate traffic behavior by learning from a realistic traffic dataset. The majority of data-driven approaches can produce only static snapshots of scenarios. How to achieve interactive simulations remains a challenge. Another area of focus is on simulating realistic driving behaviors and trajectories [6, 7, 31, 37, 41]. Some studies focus on generating safety-critical situations and designing paths that induce misbehavior in autonomous vehicles [9, 38, 46], but these simulators can not cover the full spectrum of realistic traffic scenarios, and how to model competitive interactions under different traffic scenarios remains a critical challenge.

3 Problem Formulation

Decentralized Partially Observable MDP (Dec-POMDP). We formulate the competitive game into a Dec-POMDP $(\mathcal{S}, \{\Omega_i, \mathcal{A}_i, \mathcal{O}_i, r_i\}_{i=1}^{I}, p_{\mathcal{T}}, \gamma, \mu_0)$ where:

1) i runs from 1 to I denotes the number of agents.
2) Ω_i and $\mathcal{A}_i$ denote the spaces of observations and actions for a specific agent i. The observed features include **map, the historical trajectories, and neighboring agents'** information. Our action space $(\Delta x, \Delta y)$ represents changes in the agent's position, given the time step length Δt. [10].

3) $\mathcal{S}$ denotes the state space that comprises all agents' historical trajectory information, capturing the historical information of the observed features of maps and neighboring agents.
4) $\mathcal{O}_i : \mathcal{S} \to \Omega_i$ denotes the observation function that maps states to local observation for the i^{th} agent. $\boldsymbol{\mathcal{O}} = \{\mathcal{O}_1, \ldots, \mathcal{O}_I\}$ denotes the function set.
5) $r_i : \{\Omega_i \times \mathcal{A}_i\}_{i=1}^I \to \mathbb{R}^+$ denotes the agent-specific reward function that maps actions and observations from all agents to the reward of i^{th} agent.
6) $p_{\mathcal{T}} : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ [3]denotes the transition function.
7) $\gamma \in [0, 1]$ and $\mu_0 \in \Delta^{\mathcal{S}}$ denote the discount factor and the initial distribution.

General Sum Markov Games (GS-MGs). We consider the general sum Markov Game under the Dec-POMDP. For the player i, the value function $V_{i,t}^{\boldsymbol{\pi}} : \mathcal{S} \to \mathbb{R}$ and action-value function $Q_{i,t}^{\boldsymbol{\pi}} : \mathcal{S} \times \mathcal{A}_i \to \mathbb{R}$ are represented by:

$$V_{i,0}^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\mu_0, p_{\mathcal{T}}, \boldsymbol{\pi}} \left[\sum_{t=0}^{T} \gamma^t r_i(\boldsymbol{o}_t, \boldsymbol{a}_t) | \boldsymbol{o}_0 = \boldsymbol{\mathcal{O}}(s) \right] \tag{1}$$

$$Q_{i,0}^{\boldsymbol{\pi}}(s, a_i, -\boldsymbol{a}_i) = \mathbb{E}_{\mu_0, p_{\mathcal{T}}, \boldsymbol{\pi}} \left[\sum_{t=0}^{T} \gamma^t r_i(\boldsymbol{o}_t, \boldsymbol{a}_t) | \boldsymbol{o}_0 = \boldsymbol{\mathcal{O}}(s), a_{i,0} = a_i \right] \tag{2}$$

where $-\boldsymbol{a}_i = \{\mathbb{1}_{i' \neq i} a_i'\}_{i'=1}^I$ [4] denotes the joint action performed by $I-1$ players (without i'th player) and $\boldsymbol{\pi} = \{\pi_i\}_{i=1}^I$ denotes the product policy. In our framework, each player is assigned an individual reward function, denoted as $r_i(\cdot)$. This aligns with the structure of a general-sum game [28], which presents a more complex and less explored domain compared to zero-sum and cooperative games [16, 43]. Despite the inherent challenges of this approach, its adoption is essential for accurately depicting the behavior of drivers in real-world scenarios. This is because human drivers typically prioritize their objectives, and their actions inevitably affect the decision-making processes of their counterparts.

Coarse Correlated Equilibrium. In the study of GS-MGs, solutions are characterized by states of *equilibrium*, wherein the participating agents cannot further enhance their individualistic policies within the confines of the given system. Within this context, a common objective is to identify a Nash Equilibrium (NE) [30]. However, existing results [1, 2, 13, 35, 48] often focus on a two-player zero-sum setting, and identifying NE in GS-MGs is particularly challenging due to: 1) NE commonly assumes independent policies based on a full-observable state whereas our state is partially observation and depends on multiple agents' trajectories. 2) The potential for multiple NEs to exist within GS-MGs, each representing a different set of strategies agents might adopt. 2) The lack of assurance that current algorithms can efficiently discover the optimal NE or converge to any equilibrium at all.

As a result, in this study, we pivot towards a more pragmatic objective: the identification of a CCE. A CCE permits interdependencies among agents'

[3] $\Delta^{\mathcal{S}}$ denotes the probability simplex over the space $\mathcal{S}$.
[4] Throughout this work, the bold symbols (e.g., $\boldsymbol{a}$) indicate a vector of variables while the unbold ones (e.g., a) represent a single variable.

policies, allowing each agent's strategy to be informed by the strategies of others. This stands in contrast to the NE, which necessitates that each agent's policy be independently optimized. The shift in focus to CCE offers two significant advantages: 1) It more accurately reflects the decision-making processes of human drivers, who typically base their actions on the behavior of nearby vehicles. 2) It eases the challenge of convergence, given that CCEs are inherently less restrictive and more prevalent than NEs.

In alignment with this goal, unlike traditional works [29, 32] that consider the Markov policy $\pi_i^{Markov} : \mathcal{S} \to \Delta^{\mathcal{A}_i}$, our decision model further depends on historical information, including decisions and observations of other players. The corresponding general correlated policy is defined as follows.

Definition 1. *(General Correlated Policy.) At a time step t, the general correlated policy of a player $i \in [1, I]$ in our Dec-POMDP environment is defined by $\pi_i : (\{\Omega_i \times \mathcal{A}_i\}_{i=1}^{I})^{t-1} \times \Omega_i \to \Delta^{\mathcal{A}_i}$, which maps the historical states, previous behaviors of all players and the player's current observation into her current decision. The corresponding marginal policy (policies of other players) is defined by $\boldsymbol{\pi}_{-i} : (\{\Omega_i \times \mathcal{A}_i\}_{i=1}^{I})^{t-1} \times \Omega_i \to \Delta^{\boldsymbol{\mathcal{A}}_{-i}}$ where $\boldsymbol{\mathcal{A}}_{-i} = \{\mathcal{A}_1 \times \dots \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \dots \times \mathcal{A}_I\}$.*

Note that π_i depends on the decision history, and thus it is time-dependent and non-stationary. For brevity, we slightly abuse the notation by conditioning the policy on the current state s_t such that $a_{i,t} \sim \pi(\cdot|o_{i,t}, \boldsymbol{h}_t)$ where history $\boldsymbol{h} = \{(o_{i,\iota}, a_{i,\iota})\}_{\iota=0,i=1}^{t-1,I}$ records the historical actions and observation of all agents at previous $t-1$ step. Such a policy considers the historical behaviors of all players. Correspondingly, unlike the Nash equilibrium that assumes independent policies, modifying the other players' policies influences the player i's current decision in a CCE, and thus the policies from different players are correlated. This difference plays a critical role in defining the CCE in GS-MGs:

Definition 2. *(ϵ-approximate CCE in GS-MGs). A general correlated policy π (definition 1) is an ϵ-approximate Coarse Correlated Equilibrium (ϵ-CCE) if*

$$\max_{i \in [I]} \left(V_{0,i}^{\dagger, \boldsymbol{\pi}_{-i}}(s) - V_{0,i}^{\boldsymbol{\pi}}(s) \right) \leq \epsilon \tag{3}$$

where $V_{0,i}^{\dagger, \boldsymbol{\pi}_{-i}}(s) = \sup_{\pi_i'} V_{0,i}^{\pi_i', \boldsymbol{\pi}_{-i}}(s)$ denotes the best response for the i^{th} player against π_{-i}. We say $\boldsymbol{\pi}$ is an (exact) CCE if the above is satisfied with $\epsilon = 0$.

Offline MA-RL. Another significant challenge in finding the CCE of multiple driver scenarios is due to the unavailability of an interactive environment, and our algorithms must be trained with only an offline database that records the behaviors of multiple drivers simultaneously on open roads. The learned policy must be consistent with the behaviors of human drivers recorded in the dataset. Specifically, the problem can be summarized as follows:

Definition 3. *(Offline MA-RL in GS-MGs.) let $\mathcal{D}_o = \{\zeta_n, \tau_{n,1}, \dots, \tau_{n,I}\}_{n=1}^{N}$ defines the offline dataset, where $n = [N]$ defines the number of scenario, ζ_n presents the game context in the n^{th} scenario, and $\tau_{n,i} = \{o_{i,0}, a_{i,0}, \dots, o_{i,T}, a_{i,T}\}$*

denotes the trajectory of i^{th} agent in the n^{th} scenario. Given $\mathcal{D}_o$, the goal of our algorithm is to learn a $\hat{\boldsymbol{\pi}}$ with the following properties: 1) Exploitability: $\hat{\boldsymbol{\pi}}$ satisfies the ϵ-approximate CCE in definition 2, and 2) Fidelity: $\hat{\boldsymbol{\pi}}$ must be consistent with the real driver's policies such that $D_f(\hat{\boldsymbol{\pi}}, \boldsymbol{\pi}^o) \leq \xi$ where D_f and ξ denote the divergence metric (e.g., Bregman divergence in Section 4.2) and a threshold.

4 Learning Coarse Correlated Equilibrium under the Generative World Model

In this work, to solve the offline MA-RL problem in GS-MGs (definition 3), we consider a model-based RL approach that 1) trains a generative world model by imitating the trajectories in the offline dataset and 2) learns CCEs based on the predicted environment dynamics and confined actions space.

4.1 Modelling Environmental Dynamics with Generative Model

To enable MA-RL algorithms to efficiently capture ϵ-approximate CCE in GS-MGs based on an offline dataset, we introduce a generative world model [50] for representing environmental dynamics.

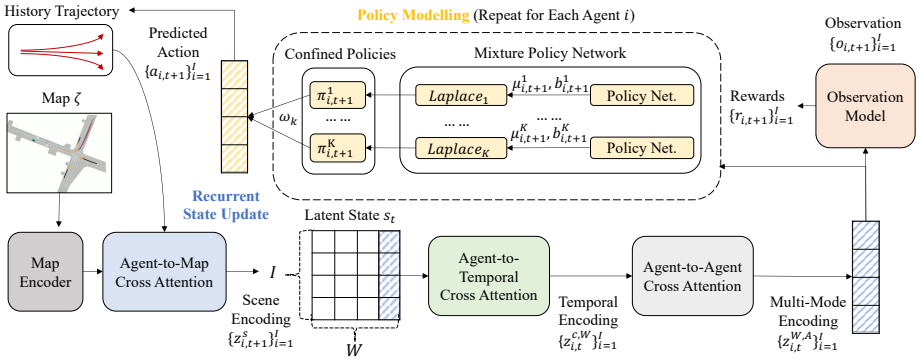

Fig. 1: Illustrating the model structure of our world model.

Model Structure. This world model (in Figure 1) including 1) *a transition model* that maps the previous state (e.g., s_{t-1}) and the action of each agent (e.g., $a_{1,t}, \ldots, a_{i,t}$) into the next state (s_t). 2) *an observation model* that maps a state s_t into agent-specific observations $o_{1,t}, \ldots, o_{I,t}$. 3) *a constrained action space* for recording candidate actions that align well with the human preference. We introduce these three components in the following: *Context-aware Transition Model.* Under the context of our Dec-POMDP, when $t = 0$, s_0 captures the static game map ζ and initial locations of all agents (or cars). when $t > 0, s_t =$

$\{\boldsymbol{z}^s_{i,w}\}^{t,I}_{w=t-W,i=0}$ (w indicates the window size of temporal features) captures the spatial-temporal information of all agents under the game map ζ in the previous $t-1$ time steps, and $a_{1,t}, \ldots, a_{i,t}$ denotes the movement of agents in the current time step. To map s_t and $\boldsymbol{a}_{1,\ldots,I,t}$ to s_{t+1}, our transition model is implemented by following parts:

- *Map Encoder* transfers the map ζ into map encoding z^c such that $z^c = \mathcal{E}(\zeta)$.
- *Agent-to-Map Cross Attention* incorporates the map information z^c into the most recent agents' trajectories $\tau_{1,t+1}, \ldots, \tau_{I,t+1}$ where $\tau_{1,t+1}$ concatenates $a_{1,t}$ and $\tau_{1,t}$ such that e.g., $\boldsymbol{z}^s_{i,t+1} = \text{CrossAttn}(\tau_{1,t+1}, \ldots, \tau_{I,t+1}, z^c)$.

The final state s_{t+1} is represented by concatenating $s_t = \{\boldsymbol{z}^s_{w,i}\}^{t,I}_{w=t-W,i=0}$ with the predicted $\boldsymbol{z}^{c,W}_{1,t+1}, \ldots, \boldsymbol{z}^{c,W}_{I,t+1}$.

Candidate Action Predictor. To align the policy well with the underlying constraint in the realistic driving scenarios, we predict the candidate actions based on the state $s_t = \{\boldsymbol{z}^s_{i,w}\}^{t,I}_{w=t-W,i=0}$ and assign a weight to each candidate actions. In specific, our action predictor is implemented by:

- *Agent-to-Temporal Cross Attention* that embeds the historical information into the current prediction, e.g., $z^W_{i,t} = \text{CrossAttn}(\{z^s_{i,t-1}, \ldots, z^s_{i,t-W}\}, z^s_{i,t})$.
- *Agent-to-Agent Cross Attention* that embed the spatial-temporal features of the surrounding agents such that $z^A_{i,t} = \text{CrossAttn}(z^W_{i,t}, \{z^W_{1,t}, \ldots, z^W_{I,t}\})$.
- *Mixture of Policy Networks.* In the k's candidate policy network, we predict the Laplace parameters $(\mu^k_{i,t}, b^k_{i,t})$ such that $(\mu^k_{i,t}, b^k_{i,t}) = f^{\text{MLP}}_{k,i}(z^A_{i,t})$.

At a time step t, $\pi_{i,t}(a_{i,t}|s_t)$ denotes the probability of the i's agent select an action a_t such that $\pi_{i,t}(a_{i,t}|s_t) = \sum_k \omega^k_i p_{\mathscr{L}}(a_t|\mu^k_{i,t}, b^k_{i,t})$ where 1) ω^k_i denotes a learnable coefficient and 2) $p_{\mathscr{L}}$ denotes the density function of the Laplace distribution. The confined policy space can be modelled as $\Pi^c_i = \{p_{\mathscr{L}}(a_t|\mu^k_{i,t}, b^k_{i,t}\}^K_{k=1}$. Due to the model outputting relative values while the input data consists of absolute values, we recalculate its anchor point each time and transform the relative position, velocity, and direction into absolute values to serve as the new input for the world model.

Agent-Specific Observation Model. Our observation model maps a state into agent-specific observations. The observation embeds the spatial-temporal features of the surrounding agents, which is implemented by $o_{i,t} = f^{\text{MLP}}_i(z^A_{i,t})$.

Training Objectives. Given a dataset recording the actions performed by each agent $\mathcal{D}_0$, at a scenario n, for each agent i, we select one of the K Laplace distributions that produces the most similar actions $\{\hat{a}^{k^*}_{i,t}\}^T_{t=0}$ to the observed ground-truth ones $\{a^o_{i,t}\}^T_{t=0}$. The supervised learning loss is denoted as:

$$\mathcal{L}_{\text{WorldModel}} = -\mathbb{E}_{\mathcal{D}_o}\Big[\underbrace{\sum_{t=0}^{T} \log\Big(p_{\mathscr{L}}(a^o_{i,t} \mid \mu^{k^*}_{i,t}, b^{k^*}_{i,t})\Big)}_{\text{bestmode\_loss}} + \underbrace{\log\Big(\omega^{k^*}_i(s_T)\Big)}_{\text{cls\_loss}}\Big] \tag{4}$$

where k^* denotes the best mode (index of the most similar Laplace), $p_{\mathscr{L}}$ denotes the Laplace density, $\mu^k_{i,t}$ and $b^k_{i,t}$ characterize the mean position and the level

of uncertainty of the i-th agent at the time step t. To automatically capture the best mode, we employ a coefficient ω_i^k denoting the probability that the k's Laplace distribution best approximates the ground-truth actions for agent i.

During the evaluation, by following [25, 50], we sample the i-th agent's future trajectory as a weighted mixture of Laplace distributions:

$$\pi^o(\hat{\tau}) = \prod_{t=1}^{\mathcal{T}} \pi_i^o(\hat{a}_{i,t} \mid o_{i,t}, \boldsymbol{h}_t) = \prod_{t=1}^{\mathcal{T}} \sum_{k=1}^{K} \omega_i^k p_{\mathscr{L}}\left(\hat{a}_{i,t} \mid \mu_{i,t}^k, b_{i,t}^k\right) \tag{5}$$

where ω_i^k can effectively act as the weighting coefficients.

4.2 Identifying CCE from Multi-player General Sum Markov Game

Since our environment is modeled as a multi-player competitive game with agent-specific rewards, inspired by [35, 36], we consider an independent update of each agent's policy where we fix the rest $I-1$ agents' policy $\boldsymbol{\pi}_{-i}$ during the update of agent i's policy π_i. In this work, we update π_i by iteratively optimizing the following objective:

$$\pi_i^j = \underset{\pi_i \in \Pi_i^c}{\arg\max}\, \mathbb{E}_{\pi_i, \mu_0}[\bar{V}_{i,t}^{\pi_i, \boldsymbol{\pi}_{-i}, j-1}(s)] - \eta_1 \mathcal{B}_\psi(\pi_i, \pi_i^o) - \frac{1}{\eta_2} \mathcal{B}_\psi(\pi_i, \pi_i^{j-1}) \tag{6}$$

where Π_i^c denotes the confined policy space predicted by the world model (Section 4.1), and the π_i^o denotes the imitation policy learned by the world model (Equation 5). This objective contains several key components that can efficiently facilitate convergence to a CCE by utilizing:

– *Optimistic value function*, which is defined by:

$$\bar{V}_{i,t}^{\pi_i, \boldsymbol{\pi}_{-i}}(s) = \mathbb{E}_{\mu_0, p_{\mathcal{T}}, \pi_i, \boldsymbol{\pi}_{-i}} \left[\sum_{\iota=t}^{T} \gamma^\iota \left[r_i^{\text{opti}}(\boldsymbol{o}_\iota, \boldsymbol{a}_\iota) \right] | \boldsymbol{o}_0 = \boldsymbol{\mathcal{O}}(s) \right] \tag{7}$$

where the optimistic reward function $r_i^{\text{opti}}(\boldsymbol{o}_\iota, \boldsymbol{a}_\iota) = r_i(\cdot) + \alpha_t(s_\iota)$ and $\alpha_t(s) = \frac{c}{m(s)}$ serves as an exploration bonus to less visited state. c is a hyper-parameter and $m(s)$ denotes the density of visited state s, representing the probability of state occurrences at time t. Such an optimistic V-learning objective serves as an extension of the CCE-V-Learning algorithm [2, 36], which is proven to converge to CCE under discrete environments, and we extend this algorithm to solve continuous decision-making problems.

– *Magnetic Mirror Descent (MMD)* [35], which is implemented by including Bregman divergence $\mathcal{B}_\psi(\cdot, \cdot)$ with respect to the mirror map ψ such that $\mathcal{B}_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle$ into the objective. In the objective (6), ρ denotes the magnet policy, π_i denotes the current policy for agent i, and π_i^{j-1} denotes the policy at previous iteration. Recent studies [5, 23, 35] justified that the mirror decent approaches can capture different kinds of equilibrium in multi-player games.

To derive a more intuitive objective, we implement the mirror map as the negative entropy such that $\psi(x) = \sum p(x) \log p(x)$, and the objective (6) becomes:

$$\pi_{i,t}^{j} = \arg\max_{\pi_i \in \Pi_i^c} \mathbb{E}_{\pi_i, \mu_0}[\bar{V}_{i,t}^{\pi_i, \boldsymbol{\pi}_{-i}, j-1}(s)] - \eta_1 \mathcal{D}_{kl}(\pi_i \| \pi_i^o) - \frac{1}{\eta_2} \mathcal{D}_{kl}(\pi_i \| \pi_i^{j-1}) \quad (8)$$

where 1) $\mathcal{D}_{kl}(\pi_i \| \pi_i^o)$ denotes the Kullback–Leibler(KL) divergence between the current policy π_i and the imitation policy, and 2) $\mathcal{D}_{kl}(\pi_i \| \pi_i^{j-1})$ denotes the KL divergence between the current policy π_i and the policy at the previous iteration π_i^{j-1}. Intuitively, by punishing the distance between current policy π_i and imitation policy π_i^o, this objective ensures the fidelity in the offline MA-RL problem (definition 3). By constraining the scale of updates, the training process becomes more stable. Since $\mathcal{D}_{kl}(\pi_i, \pi_i^o) = \mathcal{H}(\pi_i, \pi_i^o) - \mathcal{H}(\pi_i)$ and by default our objective considers the entropy of trajectory τ_i:

$$\pi_i(\tau_i) = \mu_0(s_0) \prod_{t=0}^{T-1} [p_{\mathcal{T}}(s_{t+1}|s_t, \boldsymbol{a}_t) \pi_{i,t}(a_{i,t}|o_{i,t}, \boldsymbol{h}_t) \boldsymbol{\pi}_{-i,t}(\boldsymbol{a}_{-i,t}|\boldsymbol{o}_{-i,t}, \boldsymbol{h}_t)]^{\gamma^t} \quad (9)$$

However, both the transition function $p_{\mathcal{T}}$ and policy of other players $\boldsymbol{\pi}_{-i,t}$ are not subject to optimize in the objective (8), and thus recent studies [11, 12] often consider the discounted causal entropy [52] $\sum_{t=0}^{T} \gamma^t \mathcal{H}[\pi(a_{i,t}|o_{i,t}, \boldsymbol{h}_t)]$. Similarly, instead of utilizing the computationally intractable trajectory-level KL-divergence $\mathcal{D}_{kl}(\pi_i \| \pi_i^{j-1})$, we consider the time-wise causal KL-divergence $\sum_{t=0}^{T} \gamma^t \mathcal{D}_{kl}[\pi_{i,t}(\cdot) \| \pi_{i,t}^{j-1}(\cdot)]$, and by substituting it and the equation (7) into the objective (8), we have:

$$\max_{\pi_i \in \Pi_i^c} \mathbb{E}\Big[\sum_{t=0}^{T} \gamma^t \Big(r_i^{\text{opti}}(\boldsymbol{o}_t, \boldsymbol{a}_t) - \eta_1 \mathcal{D}_{kl}[\pi_{i,t}(\cdot) \| \pi_{i,t}^o(\cdot)] - \frac{1}{\eta_2} \mathcal{D}_{kl}[\pi_{i,t}(\cdot) \| \pi_{i,t}^{j-1}(\cdot)]\Big)\Big] \quad (10)$$

where, for brevity, we denote $\pi_{i,t}(a_{i,t}|o_{i,t}, \boldsymbol{h}_t)$ as $\pi_{i,t}(\cdot)$. Since the KL-divergence of two variables (x, y) can be represented as $\mathcal{D}_{kl}(x, y) = \mathcal{H}(x, y) - \mathcal{H}(x)$, objective (10) can be further derived as:

$$\max_{\pi_i \in \Pi_i^c} \mathbb{E}\Big[\sum_{t=0}^{T} \gamma^t \Big(r_i^{\text{opti}}(\boldsymbol{o}_t, \boldsymbol{a}_t) + \eta \mathcal{H}[\pi_{i,t}(\cdot)] + \mathbb{E}_{\pi_{i,t}}[\log(\pi_{i,t}^o)^{\eta_1} (\pi_{i,t}^{j-1})^{\frac{1}{\eta_2}}]\Big)\Big] \quad (11)$$

where for brevity, we denote $\eta = \frac{1+\eta_1\eta_2}{\eta_2}$. This objective maximizes the entropy of learned policy $\pi_{i,t}$, which aligns well with the soft Bellman update, and thus we propose a soft actor-critic algorithm in the following.

An Optimistic Soft Actor-Critic Implementation. To effectively optimize the objective (11), we propose a CCE Multi-Agent Soft Actor-Critic algorithm to update the policy $\pi_{i,t}(a_{i,t}|o_{i,t}, \boldsymbol{h}_t)$ by following the CTDE framework. Algorithm 1 (see Appendix D) introduces the implementation of CCE-MASAC. To cope with the general-sum Markov game, we independently parameterize and update both the actor and critic models conditioning on each agent.

Multi-agent Soft Policy Evaluation. Since the objective (11) considers a maximum entropy policy, our policy evaluation objective utilizes the soft Bellman operator [11]. We construct two critic networks (parameterized by $\phi, \hat{\phi}$) and the corresponding loss function can be given by:

$$\mathcal{L}(\phi_{i,t}) = \mathbb{E}_{s_t, s_{t+1}, a_{i,t}, r_{i,t} \sim B}[(Q_{\phi_{i,t}}(s_t, a_{i,t}, \boldsymbol{a_{-i,t}}) - y_{i,t})^2] \tag{12}$$
$$y_{i,t} = r_{i,t}^{\text{opti}} + \log(\pi_{i,t}^{o})^{\eta_1}(\pi_{i,t}^{j-1})^{\frac{1}{\eta_2}} + \gamma \mathbb{E}_{\pi_{\theta_{i,t}}}[\eta Q_{\hat{\phi}_{i,t}}(s_t, a_{i,t}, \boldsymbol{a_{-i,t}}) - \eta \log \pi_{\theta_{i,t}}(\cdot)]$$

where θ is the policy network and B denotes a memory buffer that stores the recently generated trajectory by the learned environmental model. Under this objective, the optimal policy follows the Boltzmann representation:

$$\pi_{\theta_{i,t}}(a_{i,t}|o_{i,t}, \boldsymbol{h}_t) \propto \left[\exp(Q_{\phi_{i,t}})\right]^{\frac{1}{\eta}} \tag{13}$$

Intuitively, our policy $\pi_{\theta_{i,t}}$ should be proportional to the imitation policy $\pi_{i,t}^{o}$ and the previous policy $\pi_{i,t}^{j-1}$, and the exponential of cumulative rewards $\exp(Q_{\hat{\phi}_{t,n}}^{\min})$.
Multi-agent Soft Policy Improvement. Based on soft policy iteration [12] and our policy representation (13), our policy update loss can be defined as follows:

$$\mathcal{L}_\pi(\theta_{i,t}) = \mathbb{E}_{s_t \sim \mathcal{D}, \tilde{\epsilon}_{i,t} \sim U}\left[\eta \log(\pi_{\theta_{i,t}}^{j}(a_{i,t}^{\tilde{\epsilon}_{i,t}}|\cdot) - Q_{\phi_{i,t}}(s_t, a_{i,t}^{\tilde{\epsilon}_{i,t}}, \boldsymbol{a_{-i,t}})\right] \tag{14}$$

where U denotes a uniform distribution supported in the range [-1,1] and $a_{i,t}^{\tilde{\epsilon}_{i,t}} = \mu - b\,\text{sgn}(\epsilon_{i,t})\ln(1 - |\epsilon_{i,t}|)$ (where Laplace parameters are predicted by neural function, i.e., $\{\mu, b\} = f_{\theta_{i,t}}(o_{i,t}, \boldsymbol{h}_t)$) denotes a reparameterizated policy (Appendix C shows the derivation). We adopt this reparameterization trick [19] since our policy follows a Laplace distribution, and sample actions from the Laplace distribution are non-differentiable.

We follow [12] and perform soft policy evaluation and improvement alternately until the learned policy converges toward the optimal policy.

5 Experiments

We validate the performance of CCE-MASAC in the task of traffic simulation.
Evaluation Environment. To study how well our algorithm captures CCEs, our empirical evaluation mainly focuses on the agents' behavior under realistic scenarios that are carefully selected from the Argoverse 2 dataset [40]. Figure 2 visualizes these intriguing scenarios, including 1) *Wrong-way driving* where some cars accidentally drive on the opposite lane, 2) *Four-lane intersection* where four cars with different destinations meet simultaneously on an intersection, 3) *T-junction merging* where cars from opposite lanes intend to merge on the same lane under dense traffic, and 4) *Dense-lane intersection* where cars enter a four-way intersection with dense traffic. To provide a fair comparison these scenarios are hold-out from the pretraining dataset.
Metrics and Running Settings. Unlike standard RL algorithms that consider rewards maximization, in the *multi-agent setting*, we primarily consider the

exploitability [35], which measures the scale of an agent's policy can exploit the current policy. An ideal equilibrium should have zero exploitability, meaning no agent alone can achieve larger rewards. In this work, we follow [49] and utilize the **CCE-gap** (see definition 2) as a measurement of exploitability. For a comprehensive analysis, we report **Discounted Cumulative Rewards** to evaluate the effectiveness of a *single agent's* policy across various traffic scenarios, and we contrast this with the rewards optimization performance of multiple agents. In our experiments, we use three different random seeds to ensure the robustness of our results, and we present the mean± standard deviation (std) for each evaluated algorithm. Appendix B shows the detailed parameter configurations. You can also check Appendix E to see the training details.

Comparison Methods. Our baseline methods include: 1) **MASAC** is a multi-agent off-policy algorithm developed for Maximum Entropy RL without applying optimistic value functions or magnetic mirror descent. 2) **Multi-agent Proximal Policy Optimization (MAPPO)** [44] adapts PPO [33] for multi-agent domains to align with the CTDE structure. 3) **QCNet** [50] jointly predict the trajectory of multiple agents under a supervised learning framework. 4) **GameFormer** [15] predicts multiple agents' trajectories by applying the hierarchical game theory to model the interaction between agents.

5.1 Efficiency of Algorithms in Closing the CCE-gap

We evaluate our CCE-MASAC by how well it can close the CCE-gap when compared to other baselines. Given a learned joint policy $\boldsymbol{\pi}$, to estimate the CCE-gap for an agent i , we fix the policies of other $I-1$ agent (i.e., $\boldsymbol{\pi}_{-i}$) and find the policy $\pi_i^\dagger$ until it reach an optimal value $V_{0,i}^{\dagger,\boldsymbol{\pi}_{-i}}(s)$. According to Equation 2, the CCE-gap can be defined as $\sum_{s\in\tau}[V_{0,i}^{\dagger,\boldsymbol{\pi}_{-i}}(s) - V_{0,i}^{\boldsymbol{\pi}}(s)]$, measuring how well the optimal policy can exploit the learned policy. In our experiment, $\pi_i^\dagger$ can be computed by leveraging the prior knowledge in reward designs (detailed in Appendix A). Figure 3 and Table 2 (see Appendix F) show the CCE-gap for each agent controlled by the algorithm in every scenario. Comparing all methods comprehensively, CCE-MASAC outperforms all others by having a smaller the CCE-gap, the better). QCNet and GameFormer predict whether each vehicle's trajectory is the same as the ground truth, without considering the competitiveness between agents, so it results in a larger CCE-gap. MAPPO performs inherently unstable and struggles to converge to the CCE. Although MASAC performs better than other baselines, it still falls short of the results obtained by CCE-MASAC without considering the property of CCE.

5.2 Efficacy of the World Model in Facilitating Policy Update

To demonstrate the learned world model can significantly facilitate policy update, we first train each agent using the regular single-agent RL algorithms (e.g., SAC [12]) and imitation learning methods (e.g., QCNet [50]). This allows us to study whether the RL agent can maximize the discounted cumulative rewards

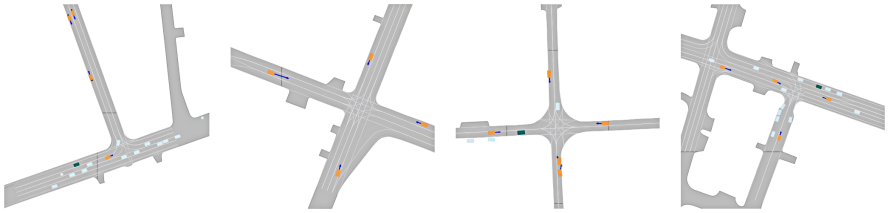

Fig. 2: From left to right, these scenarios are Wrong-way driving, Four-lane intersection, T-junction merging, and Dense-lane intersection. In each traffic scenario, only the orange vehicles are controlled by the algorithm. The gray agents represent bicycles, buses, and other types of vehicles that are not manipulated by the algorithm in the environment. The green vehicles are autonomous cars. Blue arrows indicate the orientation of each agent, while **black** represents the trajectories of the vehicles. In the first scenario, the two yellow cars at the top are violating traffic rules by driving in the opposite direction. All other vehicles in each scenario initially adhere to traffic rules by default.

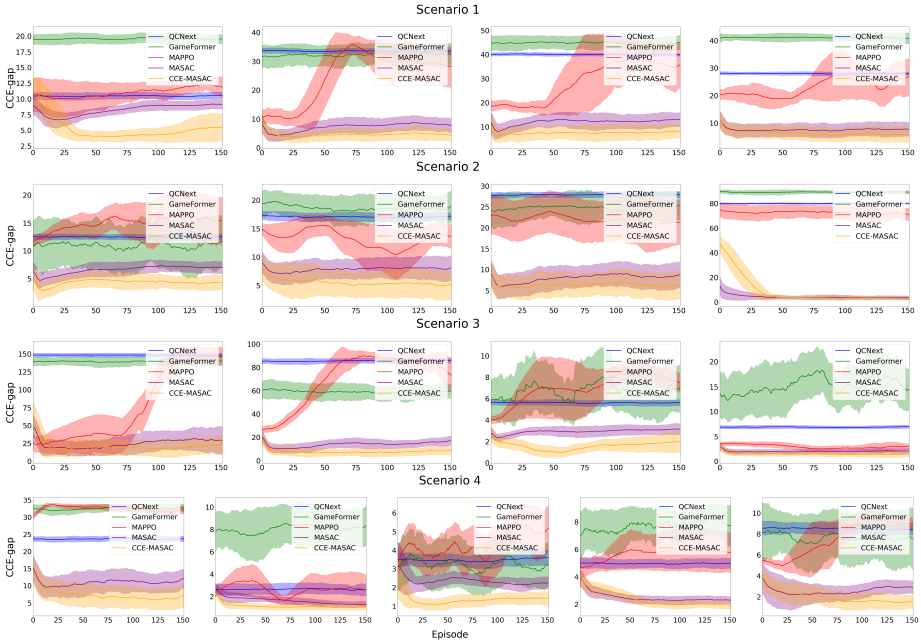

Fig. 3: The training curve of the CCE-gap across different *episodes*, where each row represents a scenario, and scenarios 1-4 correspond to Wrong-way driving, Four-lane intersection, T-junction merging, and Dense-lane intersection respectively. Each column corresponds to one of the agents in the multi-agent environment. (QCNext [51] is the multi-agent version of QCNet.)

under the learned world model. Figure 4 shows the rewards collected by each agent across various scenarios. We observed SAC can improve the agents' reward maximization performance under our world model, compared to direct imitation. Additionally, in this experiment, the results demonstrate the rewards obtained by CCE-MASAC are *lower than* those obtained by SAC, indicating that control performance can be influenced by the competitive behavior under our evaluation environments. These findings demonstrate our world model provides a valid environment for multi-agent policy updates.

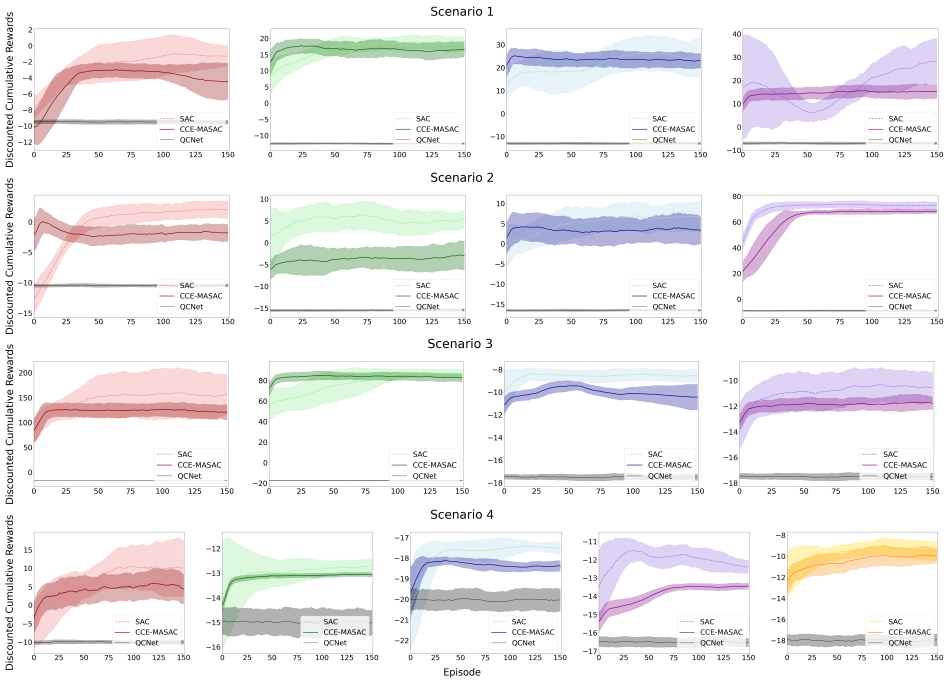

Fig. 4: The discounted cumulative rewards obtained by each agent in single-agent, multi-agent, and behavior cloning settings. Each row represents a scenario, and each column corresponds to one of the orange agents controlled by the algorithm.

5.3 Case Study: Visualization of the Learned CCEs

Figure 5 visualizes the CCE learned by CCE-MASAC in each scenario. Under the framework of general-sum games, each agent is self-interested. To maximize their rewards, these agents tend to drive quickly, resulting in competitive behaviors such as lane hogging and aggressive merging. These behaviors lead to congestion in both the straight lanes and intersections. In such situations, individual decisions are not only influenced by their own choices (whether to pass or wait) but also by

the choices of other agents. When agents strive to prioritize their interests, they may exacerbate the congestion until no agent can further improve its policy. This situation characterizes a CCE, where individual policy choices are interdependent, and no one can change their policy to gain benefits given the policies of others. These situations can effectively represent the infrequent but safe-critical events, which pose significant challenges for many autonomous driving systems.

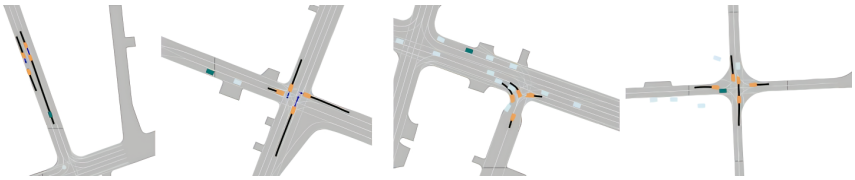

Fig. 5: We visualized the CCEs learned by CCE-MASAC across different scenarios. This figure depicts the trajectories generated by the algorithm at the final timestep.

5.4 Limitations

Omitting the Constraint: CCE-MASAC does not account for the constraints of agents, failing to model their behavior in avoiding constraint violations within competitive environments, which frequently occurs in real-world traffic situations. **Exclusion of Cooperation:** While we focus on modeling the competitive behaviors of vehicles, realistic traffic scenarios might involve both competition and cooperation behaviors among agents during some specific events (e.g., lane merging or yielding).

6 Conclusion

In this work, we introduce a generative world model aimed at predicting the environmental dynamics of self-driving scenarios. Leveraging this world model, we introduce the CCE-MASAC algorithm to effectively capture competitive behaviors among agents, thereby identifying CCE from a game-theoretical view. To exhibit the efficacy of our approach compared to other baselines, we explore the ability of CCE-MASAC to precisely estimate CCE and how well the generative model facilitates policy updates. Future research could incorporate velocity or distance constraints into the algorithm design and investigate how CCE-MASAC can be applied in mixed settings involving cooperative behaviors among vehicles.

Acknowledgements.

This work is supported in part by Shenzhen Fundamental Research Program (General Program) under grant JCYJ20230807114202005, Guangdong-Shenzhen Joint Research Fund under grant 2023A1515110617, Guangdong Basic and Applied Basic Research Foundation under grant 2024A1515012103, Shenzhen Science and Technology Program ZDSYS20211021111415025, and Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence (2023B1212010001).

References

1. Bai, Y., Jin, C., Wang, H., Xiong, C.: Sample-efficient learning of stackelberg equilibria in general-sum games. In: Advances in Neural Information Processing Systems, (NeurIPS). pp. 25799–25811 (2021)
2. Bai, Y., Jin, C., Yu, T.: Near-optimal reinforcement learning with self-play. In: Advances in Neural Information Processing Systems, (NeurIPS) (2020)
3. Bergamini, L., Ye, Y., Scheel, O., Chen, L., Hu, C., Del Pero, L., Osiński, B., Grimmett, H., Ondruska, P.: Simnet: Learning reactive self-driving simulations from real-world observations. In: IEEE International Conference on Robotics and Automation, (ICRA). pp. 5119–5125 (2021)
4. Brockfeld, E., Kühne, R.D., Skabardonis, A., Wagner, P.: Toward benchmarking of microscopic traffic flow models. Transportation Research Record **1852**, 124 – 129 (2003)
5. Cen, S., Chi, Y., Du, S.S., Xiao, L.: Faster last-iterate convergence of policy optimization in zero-sum markov games. In: International Conference on Learning Representations, (ICLR) (2023)
6. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In: Annual Conference on Robot Learning, (CoRL). vol. 100, pp. 86–99 (2019)
7. Chen, Y., Ivanovic, B., Pavone, M.: Scept: Scene-consistent, policy-based trajectory predictions for planning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR). pp. 17082–17091 (2022)
8. Dosovitskiy, A., Ros, G., Codevilla, F., López, A.M., Koltun, V.: Carla: An open urban driving simulator. In: Annual Conference on Robot Learning, (CoRL) (2017)
9. Feng, L., Li, Q., Peng, Z., Tan, S., Zhou, B.: Trafficgen: Learning to generate diverse and realistic traffic scenarios. In: IEEE International Conference on Robotics and Automation, (ICRA). pp. 3567–3575 (2023)
10. Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., Co-Reyes, J.D., Agarwal, R., Roelofs, R., Lu, Y., Montali, N., Mougin, P., Yang, Z., White, B., Faust, A., McAllister, R., Anguelov, D., Sapp, B.: Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. CoRR **abs/2310.08710** (2023)
11. Haarnoja, T., Tang, H., Abbeel, P., Levine, S.: Reinforcement learning with deep energy-based policies. In: International Conference on Machine Learning, (ICML). pp. 1352–1361 (2017)
12. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning, (ICML). vol. 80, pp. 1856–1865 (2018)

13. Hambly, B.M., Xu, R., Yang, H.: Policy gradient methods find the nash equilibrium in n-player general-sum linear-quadratic games. Journal of Machine Learning Research, (JMLR) **24**, 139:1–139:56 (2023)
14. Hu, J., Wellman, M.P.: Nash q-learning for general-sum stochastic games. Journal of Machine Learning Research, (JMLR) **4**, 1039–1069 (2003)
15. Huang, Z., Liu, H., Lv, C.: Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In: International Conference on Computer Vision, (ICCV). pp. 3880–3890 (2023)
16. Hwang, K.S., Chiou, J.Y., Chen, T.Y.: Cooperative reinforcement learning based on zero-sum games. SICE Annual Conference pp. 2973–2976 (2008)
17. Jin, C., Liu, Q., Wang, Y., Yu, T.: V-learning—a simple, efficient, decentralized algorithm for multiagent rl. In: International Conference on Learning Representations, (ICLR Workshop) (2022)
18. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. IEEE/CVF International Conference on Computer Vision, (ICCV) pp. 4550–4559 (2019)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations, (ICLR) (2014)
20. Leonardos, S., Overman, W., Panageas, I., Piliouras, G.: Global convergence of multi-agent policy gradient in markov potential games. In: International Conference on Learning Representations, (ICLR) (2022)
21. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: European Conference on Computer Vision, (ECCV). vol. 12347, pp. 541–556 (2020)
22. Lioutas, V., Scibior, A., Wood, F.: Titrated: Learned human driving behavior without infractions via amortized inference. Transactions on Machine Learning Research, (TMLR) (2022)
23. Liu, M., Ozdaglar, A.E., Yu, T., Zhang, K.: The power of regularization in solving extensive-form games. In: International Conference on Learning Representations, (ICLR) (2023)
24. Liu, S., Zhu, M.: Distributed inverse constrained reinforcement learning for multi-agent systems. In: Neural Information Processing Systems, (NeurIPS) (2022)
25. Liu, S., Zhu, M.: Learning multi-agent behaviors from distributed and streaming demonstrations. In: Neural Information Processing Systems, (NeurIPS) (2023)
26. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: IEEE International Conference on Intelligent Transportation Systems, (ITSC). pp. 2575–2582 (2018)
27. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, (NeurIPS). pp. 6379–6390 (2017)
28. Mao, W., Basar, T.: Provably efficient reinforcement learning in decentralized general-sum markov games. Dynamic Games And Applications **13**(1), 165–186 (2023)
29. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)

30. Nash, J.: Non-cooperative games. Annals of mathematics pp. 286–295 (1951)
31. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: European Conference on Computer Vision, (ECCV). vol. 12363, pp. 683–700 (2020)
32. Schulman, J., Levine, S., Abbeel, P., Jordan, M.I., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning, (ICML). vol. 37, pp. 1889–1897 (2015)
33. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. CoRR **abs/1707.06347** (2017)
34. Ścibior, A., Lioutas, V., Reda, D., Bateni, P., Wood, F.: Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation. In: IEEE International Intelligent Transportation Systems Conference, (ITSC). pp. 720–725 (2021)
35. Sokota, S., D'Orazio, R., Kolter, J.Z., Loizou, N., Lanctot, M., Mitliagkas, I., Brown, N., Kroer, C.: A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In: International Conference on Learning Representations, (ICLR) (2023)
36. Song, Z., Mei, S., Bai, Y.: When can we learn general-sum markov games with a large number of players sample-efficiently? In: International Conference on Learning Representations, (ICLR) (2022)
37. Suo, S., Regalado, S., Casas, S., Urtasun, R.: Trafficsim: Learning to simulate realistic multi-agent behaviors. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR). pp. 10400–10409 (2021)
38. Suo, S., Wong, K., Xu, J., Tu, J., Cui, A., Casas, S., Urtasun, R.: Mixsim: A hierarchical framework for mixed reality traffic simulation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR). pp. 9622–9631 (2023)
39. Tan, S., Wong, K., Wang, S., Manivasagam, S., Ren, M., Urtasun, R.: Scenegen: Learning to generate realistic traffic scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, (CVPR). pp. 892–901 (2021)
40. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Advances in Neural Information Processing Systems Track on Datasets and Benchmarks 1, (NeurIPS) (2021)
41. Xu, D., Chen, Y., Ivanovic, B., Pavone, M.: Bits: Bi-level imitation for traffic simulation. In: IEEE International Conference on Robotics and Automation, (ICRA). pp. 2929–2936. IEEE (2023)
42. Yan, X., Zou, Z., Feng, S., Zhu, H., Sun, H., Liu, H.X.: Learning naturalistic driving environment with statistical realism. Nature Communications **14**(1), 2037 (2023)
43. Yang, Y., Wang, J.: An overview of multi-agent reinforcement learning from game theoretical perspective. CoRR **abs/2011.00583** (2020)
44. Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A.M., Wu, Y.: The surprising effectiveness of PPO in cooperative multi-agent games. In: Advances in Neural Information Processing Systems, (NeurIPS) (2022)
45. Yu, Z., Yang, J., Huang, H.H.: Smoothing regression and impact measures for accidents of traffic flows. Journal of Applied Statistics **51**, 1041 – 1056 (2023)
46. Zhang, C., Tu, J., Zhang, L., Wong, K., Suo, S., Urtasun, R.: Learning realistic traffic agents in closed-loop. In: Annual Conference on Robot Learning, (CoRL) (2023)

47. Zhang, K., Koppel, A., Zhu, H., Basar, T.: Global convergence of policy gradient methods to (almost) locally optimal policies. SIAM Journal on Control and Optimization **58**(6), 3586–3612 (2020)
48. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of Reinforcement Learning and Control pp. 321–384 (2021)
49. Zhang, Y., Zhang, R., Gu, Y., Li, N.: Multi-agent reinforcement learning with reward delays. In: Learning for Dynamics and Control Conference, (L4DC). vol. 211, pp. 692–704 (2023)
50. Zhou, Z., Wang, J., Li, Y., Huang, Y.: Query-centric trajectory prediction. In: International Conference on Computer Vision and Pattern Recognition, (CVPR). pp. 17863–17873 (2023)
51. Zhou, Z., Wen, Z., Wang, J., Li, Y., Huang, Y.: Qcnext: A next-generation framework for joint multi-agent trajectory prediction. CoRR **abs/2306.10508** (2023)
52. Ziebart, B.D., Bagnell, J.A., Dey, A.K.: Modeling interaction via the principle of maximum causal entropy. In: International Conference on Machine Learning, (ICML). pp. 1255–1262 (2010)