Socially-Attentive Policy Optimization in Multi-Agent Self-Driving System

Anonymous Author(s)
Affiliation
Address
email

Abstract: As increasing numbers of autonomous vehicles (AVs) are being deployed, it is important to construct a multi-agent self-driving (MASD) system for navigating traffic flows of AVs. In an MASD system, AVs not only navigate themselves to pursue their own goals, but also interact with each other to prevent congestion or collision, especially in scenarios like intersection or lane merging. Multi-agent reinforcement learning (MARL) provides an appealing alternative to generate safe and efficient actions for multiple AVs. However, current MARL methods are limited to describe scenarios where agents interact in either a cooperative of competitive fashion within one episode. Ordinarily, the agents' objectives are defined with a global or team reward function, which fail to deal with the dynamic social preferences (SPs) and mixed motives like human drivers in traffic interactions. To this end, we propose a novel MARL method called Socially-Attentive Policy Optimization (SAPO), which incorporates: (a) a self-attention module to select the most interactive traffic participant for each AV, and (b) a social-aware integration mechanism to integrate objectives of interacting AVs by estimating the dynamic social preferences from their observations. SAPO solves the problem of how to improve the safety and efficiency of MASD systems, by enabling AVs to learn socially-compatible behaviors. Simulation experiments show that SAPO can successfully capture and utilize the variation of the SPs of AVs to achieve superior performance, compared with baselines in MASD scenarios.

Keywords: self-driving, social preference, multi-agent, reinforcement learning

1 Introduction

With the development of self-driving technologies, inreasing numbers of autonomous vehicles (AVs) are being deployed on public roads, and sharing the space with other traffic participants. Different from the single-agent platforms [1] built for realism in rendering a single AV under different road and weather conditions, the Multi-Agent Self-Driving (MASD) system [2, 3, 4] is designed to navigate traffic flows of multiple AVs as self-driven particles, where AVs need to pursue their own goals and interacts with each other in vehicle interaction events (e.g., intersection and lane merging). As with human drivers and pedestrians who usually show respect to others [5], it is of vital importance for AVs to consider how to form socially-compatible behaviors by considering different social preferences (SPs) when making decisions, as opposed to simply optimizing the passive response to any sudden actions of the surrounding entities [6]. Fig. 1 shows the illustrative examples where interacting AVs behave differently in different SPs. Failing to incorporate an awareness of other agents' SPs can lead to increased congestion as well as more collisions between vehicles. Thus, an efficient MASD system needs AVs to generate socially-compatible behaviors that are not only optimal in terms of their own interests, but must also have an awareness of other AVs and their intentions.

Enabling this produces a series of challenges: First, for the efficiency and accuracy of modeling SPs, it is necessary to determine which ranges or groups of AVs are actually interacting. For example,

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

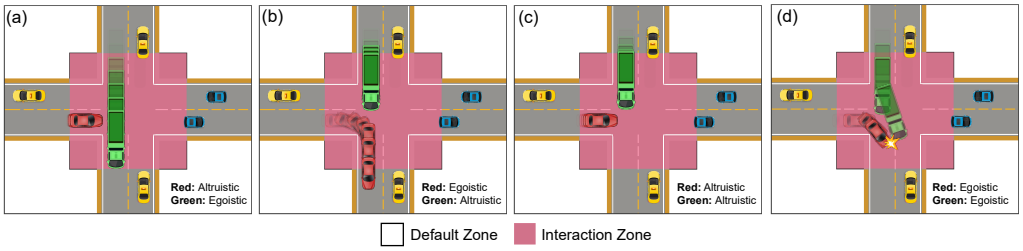


Figure 1: Examples of an MASD system, where 2 AVs are in different social preferences.

as shown in Fig. 1, most of the interaction events occur in the red zone when AVs are crossing the intersection together. Second, as time goes by, the SP of an AV usually varies because it may play various roles when interacting with other AVs in different kinds or driving styles.

To this end, the goal of this paper is to capture the most likely object which each AV interacts with, and then extract their SPs to improve the efficiency of MASD control algorithms. Existing motion planning models [7, 8, 9] and safe control methods [10, 11] only focus on the physical states and subjective observations of the ego car, but fail to consider socially-compatible behaviors. Multi-agent reinforcement learning (MARL) is becoming increasingly popular for several sequential decision-making scenarios [12, 13, 14] with multiple controllable agents. Exemplar methods like QMIX [15] and MADDPG [12] are widely used in MASD systems with multiple controllable agents in fully cooperative or fully competitive settings, by defining a shared team reward or individual payoffs as the objectives. However, they use specific reward functions (with fixed parameters of different factors) to evaluate the behaviours, which fail to consider the dynamic SPs of AVs.

Contributions. Our main contribution is a novel MARL method called Socially-Attentive Policy Optimization (SAPO) for independent AV control, which solves the problem of how to improve the safety and efficiency of an MASD system by learning socially-compatible behaviors. This method consists of two key components, (i) a self-attention module to select the most interactive participant for each AV, which extracts the critical entities in vehicle interaction events, (ii) a social-aware integration mechanism to integrate individual objectives of interacting AVs by estimating the dynamic SPs from their observations. Based on a simulation platform called SMARTS [4], we validate SAPO in several MASD scenarios, and then we showcase how SAPO navigates each AV to select the interacting object and estimate the SPs in real time, which current MARL methods cannot handle.

2 Related Work

2.1 Deep Reinforcement Learning in MASD Systems

MARL is widely used to navigate AVs in MASD systems. Wu *et al.* in [16] proposed an MARL-based velocity maximization framework by effectively leveraging the structure of the human driving behavior to form an efficient vehicle spacing for an intersection network. Isele *et al.* in [17] designed several exploratory actions (e.g., creeping behaviors) for MARL to optimize the acceleration profile along the path while safely avoiding collisions. Yang *et al.* in [18] proposed a credit-assignment method for interactions between actions and goals of different AVs in traffic simulators. Guan *et al.* in [19] proposed a model-based MARL framework to achieve centralized coordination. Peng *et al.* in [20] considered the fact that each individual agent is mostly affected by its nearby agents, and facilitate the coordination of AVs at both local and global levels in MASD systems. However, most of the existing methods only focus on how to achieve safe response to sudden actions of the surrounding entites, or just assume that each AV has a fixed SP in an episode [20], which are limited to construct the socially-compatible behaviours of different AVs reasonably in general cases.

2.2 Social Preference in Vehicle Interaction

SP is increasingly discussed to better formulate interaction between AVs in self-driving scenarios. Sun *et al.* in [21] proposed the human drivers' behavior generation model with cumulative prospect

theory to capture irrational decisions. Wang *et al.* in [22] established an online algorithm to estimate social-aware reward functions, by inferring three kind of characteristics, including egoism, courtesy and confidence. Schwarting *et al.* in [23] formulated interactions between AVs as a best-response (BR) game, and then they proposed a metric called Social Value Orientation (SVO) which quantifies the continuous degree of an agent's selfishness or altruism in reward definition. This approach allows AVs to observe other traffic participants, estimate their SVOs, and generate an autonomous control policy in real time. Furthermore, estimating dynamic SVOs at consecutive timesteps can better predict how AVs will interact and cooperate with others in MASD systems, which also inspires several solutions for several sequential social dilemma (SSD) games [24, 25]. However, existing methods focus on all of traffic participants simultaneously, which may bring increasing challenges as deploying more vehicles. Thus, in this paper, we consider how to filter redundant information by selecting the most interactive traffic participant as the interacting object for each AV.

3 Preliminaries

3.1 Dec-POMDP

We consider an MASD system where M independent AVs are self-driven particles, aiming to reach their given destinations quickly and safely. To apply MARL, we formulate this system as a decentralized partially observable Markov decision process (Dec-POMDP [26]), represented by a tuple $(\mathcal{M}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{U}, \Pr, \gamma)$, where $\mathcal{M}$, $\mathcal{S}$, $\mathcal{O}$, $\mathcal{A}$ and $\mathcal{U}$ are the set of AVs, states, local observations, actions and policies, and $\gamma \in [0, 1]$ is the discount factor. The system works as follows, episodically. At the beginning of an episode, the global state is initialized as $\boldsymbol{s}_0$. At each timestep t, each AV $m \in \mathcal{M}$ has its own observation $\boldsymbol{o}_t^m$ of state $\boldsymbol{s}_t$, including kinematic information of itself and other observable AVs. Then, it decides an action $\boldsymbol{a}_t^m$ sampled from its policy $\boldsymbol{a}_t^m \sim \pi^m(\cdot|\boldsymbol{o}_t^m)$. After the environment receives all AVs' joint action (i.e., $\mathbf{a}_t = \{\boldsymbol{a}_t^m\}_{m=1}^M$), each AV m receives the individual extrinsic reward $r_t^m = \mathcal{R}^m(\boldsymbol{o}_t^m, \boldsymbol{a}_t^m)$, which evaluates its performance at timestep t; followed by transiting to the next state $\boldsymbol{s}_{t+1}$, based on the state transition distribution $\Pr(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The episode will be terminated if: (a) all of the AVs have arrived at their destinations, (b) some of them collide with each others. For each AV m, the optimization problem is to find an optimal policy $\pi^m \in \mathcal{U}$ to maximize the accumulated discounted reward, as:

$$\begin{aligned} \max_{\pi^m} \quad & \mathbb{E}\big[\textstyle\sum_{t=1}^{\infty} \gamma^t \, r_t^m(\boldsymbol{o}_t^m, \boldsymbol{a}_t^m)\big] \\ \text{s.t.} \quad & \boldsymbol{o}_t^m \in \mathcal{O}, \ \boldsymbol{a}_t^m \in \mathcal{A}, \ \pi^m \in \mathcal{U}. \end{aligned} \tag{1}$$

Note that each AV $m \in \mathcal{M}$ optimizes its policy π^m independently in our formulated Dec-POMDP.

3.2 Independent Control by Proximal Policy Optimization

To solve this Dec-POMDP, a simple approach is the Independent Proximal Policy Optimization (IPPO [27]), where each AV maximizes individual objective as in the single-agent problems. The difference is that each AV only utilizes its observation $\boldsymbol{o}_t^m$ instead of the full state $\boldsymbol{s}_t$. Extended by the policy-based method PPO [28], the value function and advantage function of AV m at timestep t can be computed as:

$$V_t^m = V^m(\boldsymbol{o}_t^m) = \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^t r_t^m\Big], \tag{2}$$

$$A^m(\boldsymbol{o}_t^m, \boldsymbol{a}_t^m, r_t^m) := r_t^m + \gamma V_{t+1}^m - V_t^m, \tag{3}$$

Then, by adopting a truncated importance sampling factor $\rho = \frac{\pi_{\text{new}}^m(\boldsymbol{a}^m|\boldsymbol{o}^m)}{\pi_{\text{old}}^m(\boldsymbol{a}^m|\boldsymbol{o}^m)}$, the surrogate objective of IPPO is written as:

$$J_{\text{IPPO}}(\pi^m) = \mathbb{E}[\min(\rho A^m, \text{clip}(\varrho, 1-\epsilon, 1+\epsilon)A^m)], \tag{4}$$

where clip function helps restrict the incentive of optimizing π^m towards r_t^m within the interval $[1-\epsilon, 1+\epsilon]$. Note that ϵ is a real number and specify the interval. The value function V^m is updated by mean squared error loss function:

$$\mathcal{L}^m = \mathbb{E}[r_t^m + \gamma V^m(\boldsymbol{o}_{t+1}^m) - V^m(\boldsymbol{o}_t^m)]^2. \tag{5}$$

4 Methodology

We propose a novel MARL algorithm called Socially-Attentive Policy Optimization (SAPO) to learn socially-compatible behaviors and achieve better cooperative patterns in MASD systems. As shown in Fig. 2, SAPO uses IPPO as the start point of the design, and consists of two improvements: a self-attention module called "Interactive Attention" to select the most interactive participant for each AV from complex traffic flows (see Section 4.1), and a Social-aware Integration mechanism to model the impacts of dynamic SPs in vehicle interaction (see Section 4.2). For simplicity, we omit the timestep index t in this section. The whole structure is depicted by the following equations (given $\forall m \in \mathcal{M}$):

$$(p, \boldsymbol{h}^m) = \text{InteractiveAttention}(\boldsymbol{o}^m), \quad p \in \mathcal{M}, \tag{6a}$$

$$r^m_{\text{SP}} = \text{Integration}(r^m, r^p | \boldsymbol{o}^m, \boldsymbol{h}^m) \tag{6b}$$

$$\pi^m(\boldsymbol{a}^m | \boldsymbol{o}^m, \boldsymbol{h}^m) \equiv \text{f}^\pi(\boldsymbol{o}^m, \boldsymbol{h}^m), \tag{6c}$$

$$V^m(\boldsymbol{o}^m, \boldsymbol{h}^m) \equiv \text{f}^V(\boldsymbol{o}^m, \boldsymbol{h}^m), \tag{6d}$$

where p is the index of the traffic participant which is most likely to interact with AV m; r^m_{SP} represents the SP-aware reward integrated from interacting AVs; f^π, f^V are multi-layer perceptrons (MLPs). For each AV m, SAPO is running as follows. First, SAPO utilizes Interactive Attention to select the most interactive participant p and generate a compact feature $\boldsymbol{h}^m \in \mathbb{R}^n$ (see Eqn. (6a)). Then, the individual reward r^m is replaced by an SP-aware reward r^m_{SP}, through Social-aware Integration mechanism (see Eqn. (6b)). Next, through $\text{f}^\pi(\cdot)$ and $\text{f}^V(\cdot)$, the compact feature is mapped into an action distribution π^m and a value function V^m, respectively (see Eqn. (6c) and Eqn. (6d)). Finally, similar to IPPO, the advantage function of AV m is computed by Eqn. (3), for PPO updates by Eqn. (4) and Eqn. (5). Details of running SAPO are deferred in Appendix A.

4.1 Interactive Attention for Selecting Interacting Objects in Traffic Flows

Since there are usually numerous AVs in MASD system, it is necessary for each AV to filter the observed information and consider what is relevant for decision. We focus on the fact that each AV is mostly affected by its nearby agents rather than focus on all of the others simultaneously [29]. That is, each AV should pay attention to the most interactive vehicle that is close or conflict with the planned route. Inspired by [30], we propose a self-attention architecture called "interaction attention". It is used to both select the most interactive participant and generate a compact feature.

As shown in Fig. 2, for each AV m, SAPO splits $\boldsymbol{o}^m$ into the observation of the ego vehicle and the others at first, which are embedded into M vectors $\boldsymbol{x}$ of size d_x, by a linear encoder f^e. Note that each AV has its own observable zone and the feature vectors of AVs outside this zone will be set to zeros. Through MLP f^q, f^k and f^v, all of the M vectors are embedded into the queries ($\boldsymbol{q}$), keys ($\boldsymbol{k}$) and values ($\boldsymbol{v}$) of size d_k. They are then fed to the multi-head attention layer [31], composed of several heads stacked together. The difference is that we only use a single query $\mathbf{Q} = [\boldsymbol{q}^m] \in \mathbb{R}^{1 \times d_k}$ by features of the ego AV, which is compared to all of keys $\mathbf{K} = [\boldsymbol{k}^1, \cdots, \boldsymbol{k}^M] \in \mathbb{R}^{M \times d_k}$, containing observed features of each AV. The similarity between the query $\boldsymbol{q}^m$ and any AV's key $\boldsymbol{k}$ is accessed by their dot product $\boldsymbol{q}^m \boldsymbol{k}^\top$ and then scaled by $\frac{1}{\sqrt{d_k}}$. Next, in order to get the most interactive participant as the interacting object, we transform the attention weights into the one-hot vector by utilizing a Gumbel Softmax [32] function $\sigma(\cdot)$. With this hard attention scheme, the index of the only non-zero element corresponds to the index of an interactive AV. We utilize the one-hot vector to gather all of the values $\mathbf{V} = [\boldsymbol{v}^1, \cdots, \boldsymbol{v}^M] \in \mathbb{R}^{M \times d_k}$ and get the output. Overall, the attention computation for each head can be written as:

$$\text{output} = \underbrace{\sigma\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)}_{\text{attention matrix}} \mathbf{V}. \tag{7}$$

Finally, the outputs from all heads are combined, generating compact features. Without loss of features, we get the final social-aware feature $\boldsymbol{h}^m \in \mathbb{R}^n$, by adding resulting tensor of Interactive

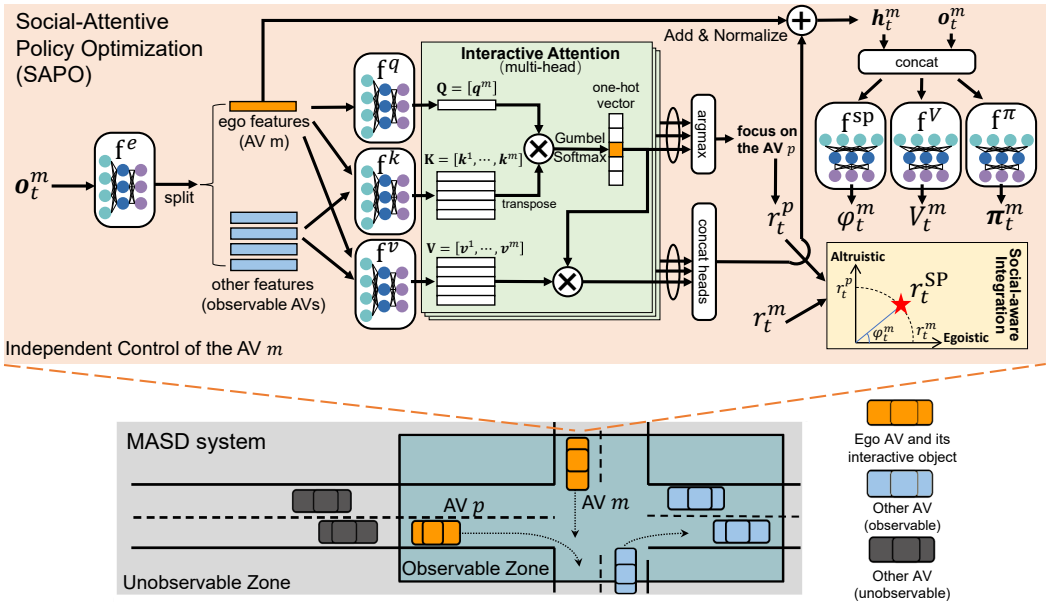


Figure 2: Architecture of SAPO.

Attention to the ego encoded observation of AV m as in residual networks. Similarly, we also combine the attention matrix of all heads to get the most interactive participant p which the AV m should focus on temporally. The advantages of the Interaction Attention are: (a) inputs can have a variable size; and (b) outputs are permutation invariant which do not depend on the ordering of surrounding AVs.

4.2 Social-Aware Integration of Individual Objectives with Dynamic Social Preferences

In an MASD system, each AV has its individual objective. However, if we simply maximize each individual reward by IPPO, the system will have sub-optimal solutions where, for example, AVs will become aggressive and egocentric, jeopardizing the performance of the population and leading to critical failures. On the contrary, if we apply fully cooperative learning schemes [15, 33] which consider the summation of individual reward as the joint objective, the trained agents will exhibit unreasonable behaviors such as sacrificing oneself to improve group reward, which is not expected in MASD systems. To find a balance, we consider utilizing the impacts of SP on learning socially-compatible behaviours. Thus, inspired by social psychology, we propose a social-aware integration mechanism to integrate objectives of interacting AVs by their SPs. Different from the previous Local Coordination Factor [20], we estimate the dynamic SP φ^m from each AV m's observation $\boldsymbol{o}^m$ at each timestep t.

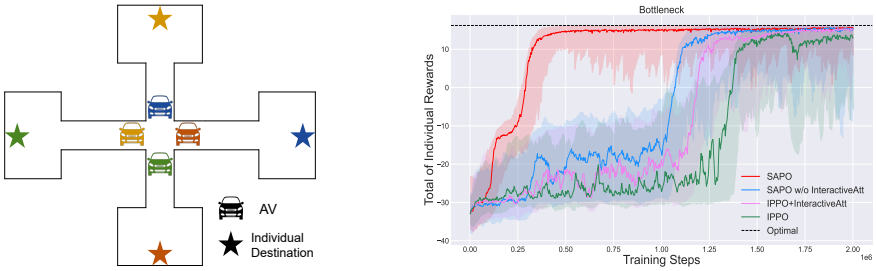
For each AV m, the mechanism works as follows, as shown in the right part of Fig. 2. Since social-aware features $\boldsymbol{h}^m$ are extracted by Interactive Attention, an MLP f^{SP} is then utilized to generate the ring measure of SVO [23] to represent the AV m's SP φ^m, as:

$$\varphi^m(\boldsymbol{o}^m, \boldsymbol{h}^m) \equiv \mathrm{f}^{\mathrm{SP}}(\boldsymbol{o}^m, \boldsymbol{h}^m); \quad \varphi^m : \mathcal{O} \times \mathbb{R}^n \mapsto [-\pi/2, \pi/2]. \tag{8}$$

Next, we utilize φ^m to control the appropriate level of coordination between the AV m and its interactive participant p, by calculating a social-aware reward r^m_{SP} as:

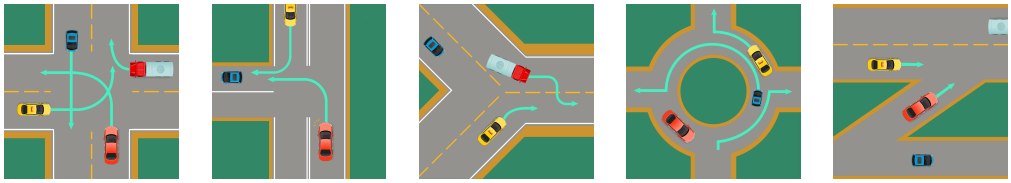
$$r^m_{\mathrm{SP}} = r^m \cos\varphi^m(\boldsymbol{o}^m, \boldsymbol{h}^m) + r^p \sin\varphi^m(\boldsymbol{o}^m, \boldsymbol{h}^m), \tag{9}$$

which replaces the AV m's individual reward r^m as a new objective. Different from automatic search of the optimal SP fixed in an episode [20], SAPO can estimate φ^m at each timestep t and make sure that r^m_{SP} is corresponding to the latest traffic conditions, which is important in real scenarios.



(a) Bottleneck (b) Learning curves of different methods

Figure 3: (a) A special toy example called Bottleneck, and each AV aims to reach the destination point with the same color as soon as possible. (b) The learning curves of SAPO and other baselines.



(a) Cross-road (b) T-junction (c) Lane Merging (d) Roundabout (e) On-ramp

Figure 4: Examples of MASD scenarios in SMARTS [4].

5 Experiments

In this section, we first study didactic examples inspired by SSD games [34] to investigate the effects of Interactive Attention and Social-aware Integration on the learning process (see Section 5.1). To demonstrate scalability on complex MASD systems, we also evaluate the performance of SAPO on a range of SMARTS benchmark tasks [4] (see Section 5.2). All experiment results are illustrated with the median performance and $25\% - 75\%$ percentiles over 5 random seeds. We test the ability of SAPO to learn reasonable socially-compatible behaviors and explore the optimal policy.

5.1 Toy Example: Bottleneck

Inspired by [25, 34], we design a toy example of MASD systems, called Bottleneck. Consider a grid world with 4 AVs (see Fig. 3(a)), 5 available actions (i.e., stop or move left, right, up and down) and observation space containing 2-D position of ego AV and the others. Each AV's reward is determined by the distance to its own destination and the time costs. Given the initial positions and destinations, AVs should learn to make ways for others and reach the goals in orders. Since the solution space is limited, we can easily explore the overall optimal policy and examine the exploitation of MARL. From Fig 3(b), we observe that SAPO trains much faster than IPPO and achieves the optimality. This is because AVs only see others as parts of the environment in IPPO. They are likely to learn selfish behaviours (e.g., keeping closer to the goal without yielding any space) and resulting in congestion, which need to consume more time on exploring the optimal policy. Besides, the learning process will be much slower without Interactive Attention. This is because it is hard to maintain the efficiency and accuracy of estimating SPs when considering all of AVs together.

5.2 SMARTS

To tackle settings that more closely represent real world settings, we focus on MASD scenarios based on the Scalable Multi-Agent RL Training School (SMARTS) benchmark. Fig. 4 describes the examples of tasks we use in our experiments. The experimental settings are deferred to Appendix B.

5.2.1 Ablation Study

To show the importance of utilizing Interactive Attention and modeling dynamic SPs on optimizing MASD systems, we perform the ablation study by: (a) removing Interactive Attention, or (b) using

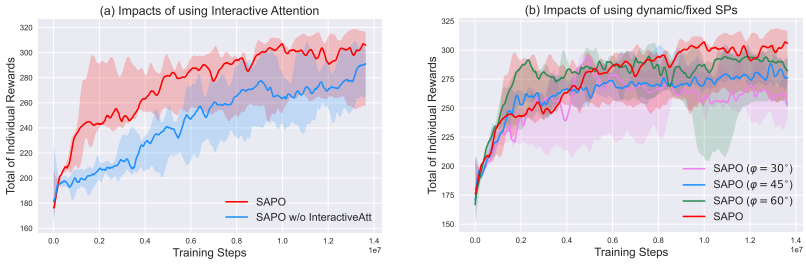


Figure 5: Ablation study.

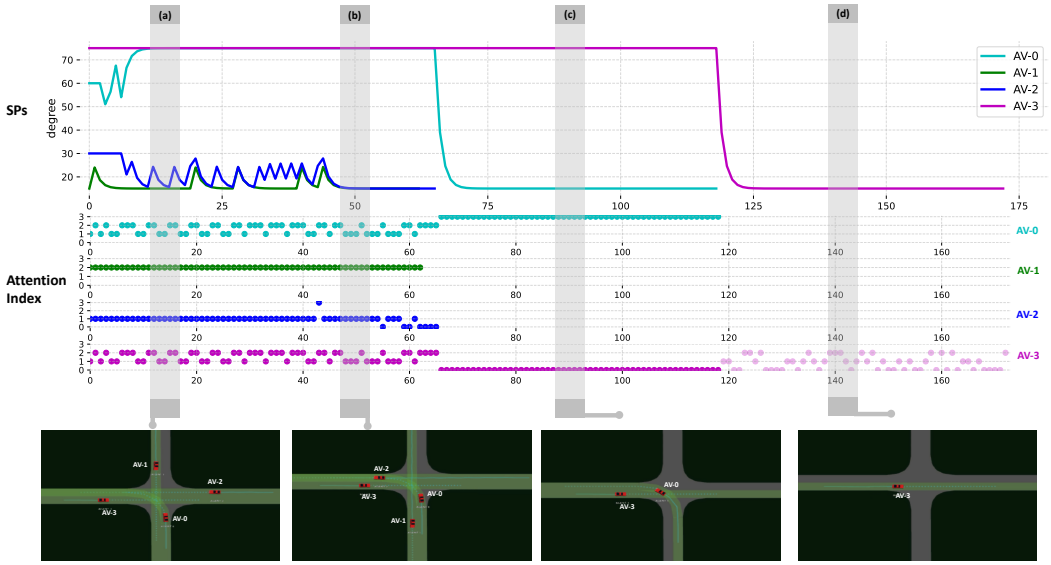


Figure 6: Illustrative visualization of AV's behaviours trained by SAPO.

given SPs instead of dynamic estimation for training SAPO in a cross-road scenario. Results are shown in Fig. 5. We see that the complete version SAPO achieves more than 5% higher rewards than SAPO w/o InteractiveAtt. This is because Interactive Attention can filter information from a number of AVs in the intersection, which is beneficial for finding the optimal policy. On the other hand, we see that SAPO ($\varphi = 30°$) performs worse than SAPO ($\varphi = 45°$) and SAPO ($\varphi = 60°$). This is because SAPO ($\varphi = 30°$) encourages more self-interested AVs (according to Eqn. (9)), which usually get blocked as they refuse to make way for others. However, even if we set $\varphi = 45°$ or $\varphi = 60°$ to keep the AVs prosocial or altruistic, SAPO is still trapped into another local optimal. We observe that only the completeness of dynamic SPs can navigate AVs to learn reasonable socially-compatible behaviours, which improves the overall performance of the MASD system.

5.2.2 Illustrative Visualization of the Trained Behaviors

In Fig. 6, we show the visualization results of AVs' positions, attention weights and SPs (from the bottom to the top) at each timestep in a cross-road scenario, where each AV follows its own waypoints, navigated by SAPO independently. To better explain the learned socially-compatible behaviors, we focus on three stages in an episode. In stage $a \rightarrow b$, AV-0 and AV-3 slow down and make way for AV-1 and AV-2 to cross the intersection. Then, in stage $b \rightarrow c$, AV-0 speeds up while AV-3 stops and waits. Finally, AV-3 speeds up and crosses the road in stage $c \rightarrow d$. From the learned SPs of each AV, we observe that AVs with higher SPs can learn to make way for AVs with lower SPs. Furthermore, we observe that Interaction Attention is helpful by selecting the most relevant traffic participant. For example, when AV-0 and AV-3 both slow down, AV-1 and AV-2 only pay attention to each other when crossing the intersection. These results prove that AVs can choose suitable vehicles as interacting objects and then learn reasonable socially-compatible behaviors.

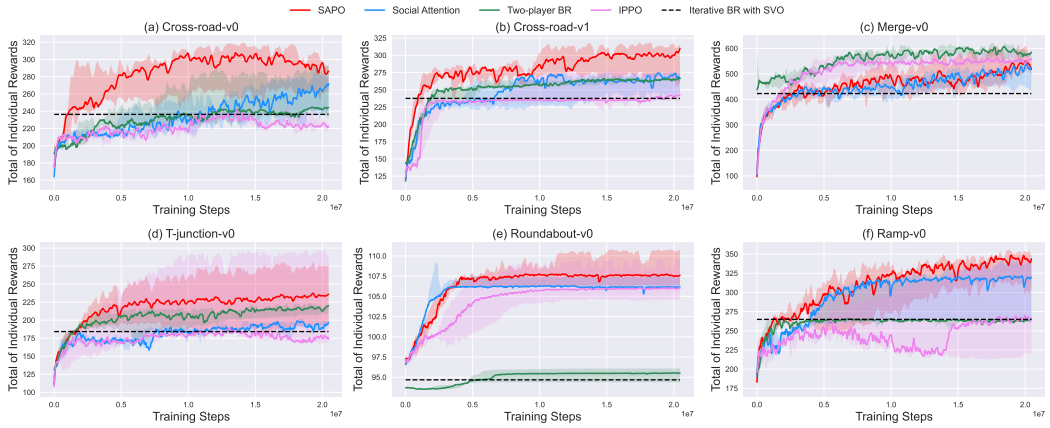


Figure 7: Training curves of SAPO and baselines in SMARTS.

5.2.3 Comparison Results of Training in MASD Scenarios

We evaluate SAPO in 6 popular MASD tasks. Other than IPPO, we further compare SAPO against three popular baselines in self-driving scenarios as follows:

- *Iterative Best Response with SVO* [23]. As a popular game-theoretic approach for self-driving, it models the interactions between AVs as a multi-player BR game wherein each AV negotiates to maximize its own reward and predict other AVs' behaviors by estimating their SVOs in real time. The details of BR and SVO estimation are deferred to Appendix D.
- *Two-player Best Response* [6]. It trains the Interactive Attention to select the interacting object for each AV, and then finds the Nash equilibrium as in two-player BR games.
- *Social Attention* [30]. Based on the centralized control of multiple AVs, it directly utilizes an attention-based architecture for decision-making involving social interactions.

To demonstrate the overall performance of each algorithm, we use the total of individual rewards as the metric. The learning curves are shown in Fig. 7. Note that we use the testing performance of Iterative BR with SVO [23] as a reference line, since it is an online optimization-based method. We observe that, SAPO significantly outperforms baselines in most of the cases and IPPO performs the worst. With attention-based feature extraction, Two-player BR and Social Attention both achieve reasonable performance, especially in lane merging scenarios where two AVs can choose when to merge so their conflicts are less than other scenarios. However, in two cross-road scenarios where AVs need to deal with more conflicts of individual rewards, the performance gap between SAPO and other baselines exceeds 5% in rewards. This proves that Social-aware Integration is important in learning socially-compatible behaviors and improving the efficiency of MASD systems.

6 Conclusion

Inspired by human cooperation in societies, this paper proposes a novel MARL method called SAPO to learn social-aware controller of MASD systems, which incorporates a self-attention module and social-aware integration mechanism for selecting interacting objects and integrating individual objectives of AVs, respectively. Experiments on basic traffic simulation show that SAPO can successfully estimate the dynamic SPs from AVs' observations and learns reasonable socially-compatible behaviours, which achieves better performance than baselines in several popular MASD systems.

7 Limitations and Future Research Directions

This paper focus on decision-making problems, so we simplify the perception of AVs and assume the observation is based on accurate sensory data, which is not easy to achieve in reality. Besides, we adopts only 6 typical MASD scenarios in a simulator as the testbed. In the future we will import High-Definition (HD) maps as AVs' observations and create more challenging scenarios for testing.

References

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun. CARLA: an open urban driving simulator. In *CoRL'17*, volume 78, pages 1–16, 2017.

[2] P. Á. López, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner. Microscopic traffic simulation using SUMO. In *itsc'18*, pages 2575–2582, 2018.

[3] Q. Li, Z. Peng, Z. Xue, Q. Zhang, and B. Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *arXiv preprint arXiv:2109.12674*, 2021.

[4] M. Zhou, J. Luo, J. Villella, Y. Yang, et al. Smarts: An open-source scalable multi-agent rl training school for autonomous driving. In *CoRL'21*, volume 155, pages 264–285, 2021.

[5] H. Wang, W. Wang, S. Yuan, X. Li, and L. Sun. On social interactions of merging behaviors at highway on-ramps in congested traffic. *CoRR*, abs/2008.06156, 2020.

[6] T. Başar and G. J. Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[7] W. Xu, J. Pan, J. Wei, and J. M. Dolan. Motion planning under uncertainty for on-road autonomous driving. In *IEEE ICRA'14*, pages 2507–2512, 2014.

[8] D. González, J. Pérez, V. M. Montero, and F. Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.*, 17(4):1135–1145, 2016.

[9] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.*, 1(1):33–55, 2016.

[10] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars. *CoRR*, abs/1708.06374, 2017.

[11] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou. Robust sampling based model predictive control with sparse objective information. In *Robotics: Science and Systems XIV, 2018*, 2018.

[12] R. Lowe, Y. I. Wu, A. Tamar, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. *NeurIPS'17*, 30, 2017.

[13] M. Samvelyan, T. Rashid, C. S. De Witt, et al. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[14] K. Kurach, A. Raichuk, P. Stańczyk, et al. Google research football: A novel reinforcement learning environment. *arXiv preprint arXiv:1907.11180*, 2019.

[15] T. Rashid, M. Samvelyan, C. Schroeder, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML'18*, pages 4295–4304, 2018.

[16] C. Wu, A. Kreidieh, E. Vinitsky, and A. M. Bayen. Emergent behaviors in mixed-autonomy traffic. In *CoRL'17*, volume 78, pages 398–407, 2017.

[17] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *IEEE ICRA'18*, pages 2034–2039, 2018.

[18] J. Yang, A. Nakhaei, D. Isele, K. Fujimura, and H. Zha. CM3: cooperative multi-goal multi-stage multi-agent reinforcement learning. In *ICLR'20*, 2020.

[19] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li. Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. *IEEE Trans. Veh. Technol.*, 69(11):12597–12608, 2020.

[20] Z. Peng, Q. Li, K. Hui, C. Liu, and B. Zhou. Learning to simulate self-driven particles system with coordinated policy optimization. In *NeurIPS'21*, pages 10784–10797, 2021.

[21] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka. Interpretable modelling of driving behaviors in interactive driving scenarios based on cumulative prospect theory. In *IEEE ITSC'19*, pages 4329–4335, 2019.

[22] L. Wang, L. Sun, M. Tomizuka, and W. Zhan. Socially-compatible behavior design of autonomous vehicles with verification on real human data. *IEEE Robotics Autom. Lett.*, 6(2): 3421–3428, 2021.

[23] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus. Social behavior for autonomous vehicles. *Proc. Natl. Acad. Sci. USA*, 116(50):24972–24978, 2019.

[24] B. Baker. Emergent reciprocity and team formation from randomized uncertain social preferences. In *NeurIPS'20*, 2020.

[25] K. R. McKee, I. M. Gemp, B. McWilliams, E. A. Duéñez-Guzmán, E. Hughes, and J. Z. Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. In *AAMAS'20*, pages 869–877, 2020.

[26] F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.

[27] C. S. de Witt, T. Gupta, D. Makoviichuk, et al. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[28] J. Schulman, F. Wolski, P. Dhariwal, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang. Mean field multi-agent reinforcement learning. volume 80, pages 5567–5576, 2018.

[30] E. Leurent and J. Mercat. Social attention for autonomous decision-making in dense traffic. *CoRR*, abs/1911.12250, 2019.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS'17*, pages 5998–6008, 2017.

[32] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *ICLR'17*. OpenReview.net, 2017.

[33] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. F. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *AAMAS'18*, pages 2085–2087, 2018.

[34] J. Z. Leibo, V. F. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *ACM AAMAS'17*, pages 464–473, 2017.