

Communication-Efficient Cooperative Multi-Agent PPO via Regulated Segment Mixture in Internet of Vehicles

Xiaoxue Yu, Rongpeng Li, Fei Wang, Chenghui Peng, Chengchao Liang, Zhifeng Zhao, and Honggang Zhang

Abstract—Multi-Agent Reinforcement Learning (MARL) has become a classic paradigm to solve diverse, intelligent control tasks like autonomous driving in Internet of Vehicles (IoV). However, the widely assumed existence of a central node to implement centralized federated learning-assisted MARL might be impractical in highly dynamic scenarios, and the excessive communication overheads possibly overwhelm the IoV system. Therefore, in this paper, we design a communication efficient cooperative MARL algorithm, named RSM-MAPPO, to reduce the communication overheads in a fully distributed architecture. In particular, RSM-MAPPO enhances the multi-agent Proximal Policy Optimization (PPO) by incorporating the idea of segment mixture and augmenting multiple model replicas from received neighboring policy segments. Afterwards, RSM-MAPPO adopts a theory-guided metric to regulate the selection of contributive replicas to guarantee the policy improvement. Finally, extensive simulations in a mixed-autonomy traffic control scenario verify the effectiveness of the RSM-MAPPO algorithm.

Index Terms—Communication-efficient, Multi-agent reinforcement learning, Regulated segment mixture, Internet of vehicles.

I. INTRODUCTION

Internet of Vehicles (IoV) emerges as an effective means to ubiquitously connect vehicles and enhance their self-driving capability (e.g., fleet management and accident avoidance). Typically, in IoV, a Connected Automated Vehicle (CAV) is contingent on Deep Reinforcement Learning (DRL) to solve diverse control tasks [1]–[3], on top of a formulated Markov Decision Process (MDP). Correspondingly, these CAVs constitute a Multi-Agent Reinforcement Learning (MARL)-empowered system. Nevertheless, the direct adoption of Independent Reinforcement Learning (IRL) [4] at the CAV, with each one accessible and responsive to a limited partial observation of the global environment, will make MARL suffer from the non-stationarity of the learning environment. Therefore, communications are generally taken into account as an indispensable ingredient in MARL [5]–[8]. For example, Ref. [5] combines Federate Learning (FL) with IRL, by regarding the aggregation of gradients as the communication,

so as to improve the involved homogeneous agents’ capability and learning efficiency. Meanwhile, individual observations [6] or intended actions [7], [8] can also be exchanged on the basis of proper encoding. Moreover, Ref. [9] proposes a stigmergy-based trustable policy collaboration scheme by directly mixing the policy parameters. But the common assumption of an existing central node in these works [5]–[9] might be impractical and underlies potential threat to the stability and timeliness of learning performance in highly dynamic scenarios like IoV. Besides, the frequent information exchange in these works inevitably generates excessive and even exponential communication overheads along with the number of agents, thus possibly overwhelming the IoV system. In a nutshell, it becomes imperative to design a communication efficient MARL algorithm.

In that regard, there has emerged intense research interest, particularly within the scope of Decentralized Federated Learning (DFL) and supervised learning. Ref. [10] puts forward a randomized selection scheme for forwarding subsets of local model parameters to their one-hop neighbors. Ref. [11] introduces a segmented gossip approach by synchronizing model segments only, thus significantly splitting the expenditure of communications. However, communication efficient system, which typically adopts a larger communication internal, faces more diverse local model updates, and may get an even worse aggregation model after simple parameter averaging [12]. Notably, this could be more exacerbated for an on-line IRL framework, since IRL agents need to interact with the environment more frequently than those for supervised learning and the processing of gradually arrived data could amplify the learning discrepancy among multiple agents. In other words, not all communicated packets will be contributive in MARL and directly adopting the over-simplistic mixture approach as in DFL works [10], [11] is far from efficiency. Instead, MARL awaits for a revolutionized mixture method and corresponding metric to regulate the aggregation of exchanged model updates, so as to ensure robust policy improvement.

In this paper, on the basis of Proximal Policy Optimization (PPO) [13], one classical policy iteration reinforcement learning algorithm, we tailor a distributed communication-efficient cooperative scheme for IRL-controlled CAVs in IoV, and propose a Regulated Segment Mixture-based Multi-Agent PPO (RSM-MAPPO) algorithm. Compared with existing communication-based MARL works, the key contributions of RSM-MAPPO can be summarized as follows.

This work was supported by the National NSF of China (62071425), the Zhejiang Key R&D Plan (2022C01093), Huawei Cooperation Project, and the Zhejiang Provincial NSF of China (LR23F010005).

X. Yu, and R. Li are with College of Information Science and Electronic Engineering, Zhejiang University (email: {sdwhyxx, lirongpeng}@zju.edu.cn). F. Wang and C. Peng are with Huawei Technologies (email: {wangfei76, pengchenghui}@huawei.com). C. Liang is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications (email: chengchaoliang@sce.carleton.ca). Z. Zhao and H. Zhang are with Zhejiang Lab as well as Zhejiang University (email: {zhaozf, honggangzhang}@zhejianglab.com).

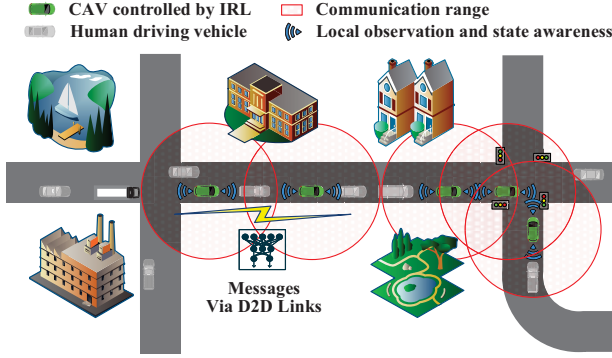


Fig. 1. Illustration of MARL in autonomous driving.

- RSM-MAPPO implements a communication-efficient MAPPO by incorporating the idea of segment mixture in DFL and augmenting multiple model replicas from received neighboring policy segments.
- In order to guarantee the policy improvement during the mixture, a theory-guided metric is developed to regulate the selection of contributive replicas only.
- Through extensive simulations in the traffic control scenario, RSM-MAPPO, which operates in a fully distributed manner, could approach the converged performance of centralized FL and IRL [5], while is significantly superior than direct application of parameters average as in DFL [10], [11], thus verifying its effectiveness.

The remainder of this paper is organized as follows. We introduce the system model and formulate the problem in Section II. Afterwards, we elaborate on the details of the proposed RSM-MAPPO algorithm in Section III. In Section IV, we present the simulation settings and discuss the experimental results. Finally, Section V concludes this paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Beforehand, we summarize the main notations in Table I.

A. System Model

As illustrated in Fig. 1, we primarily consider an IoV scenario consisting of N CAVs (i.e., PPO-empowered agents)¹ alongside some human-driving vehicles. Specifically, at each time-step t , agent i senses partial status $s_t^{(i)}$ (e.g., the speed and positions of neighboring vehicles) of the IoV environment, and then selects an action $a_t^{(i)} \in \mathcal{A}$ according to its local policy $\pi^{(i)}$ parameterized by $\theta^{(i)}$. Afterwards, an individual reward $r_t^{(i)} \in \mathcal{R}$ will be obtained, with the state transferring into $s_{t+1}^{(i)}$. Correspondingly, a sequential Markov state transition $\phi_t^{(i)} = \langle s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)} \rangle$ can be stored. In this paper, the MAPPO learning encompasses an independent local learning phase and a communication-assisted mixing phase. Generally, in the first phase, after running a policy for T time-steps (far less than the length of an epoch, which equals multiple T), we can obtain a mini-batch Φ of collected samples for iterations of local model updates. Subsequently, in the second phase, each agent i interacts with its one-hop neighbors Ω_i within

¹In this paper, we assume the terminologies “CAV” and “agent” are interchangeable.

TABLE I
MAJOR NOTATIONS USED IN THE PAPER.

Notation	Definition
$s_t^{(i)}, a_t^{(i)}, r_t^{(i)}$	Local state, individual action and reward of agent i at time step t
π, θ	Current target policy and its parameters
$\bar{\pi}, \bar{\theta}$	The referential target policy and its parameters
Ω_i	Set of one-hop neighbors within the communication range of agent i
α	Mixture metric of current parameters and referential parameters
θ_{mix}	Mixed policy parameters
p, P	Index of segments, $p = 1, 2, \dots, P$
κ	Number of model replicas
τU	Communication interval given U local iterations
v	Size of the policy parameters
ψ	Communication consumption until convergence of the IRL model

its communications range directly (e.g., via Device-to-Device (D2D) channels), so as to reduce the behavioral localities of IRL and improve their cooperation efficiency.

Algorithmically, we adopt a sample-efficient standard PPO setting in the local learning phase, which leverages two different policies (i.e., behavior policy $\pi_{\theta_{\text{old}}}$ ² for sample collection and target policy π_{θ} for online optimization) instead of the same policy in classical REINFORCE. Every U local iterations, the parameters of the target policy will be copied to those of the behavior policy. In addition, PPO implements importance sampling-based optimization using all past experiences via an adjustable ratio $\lambda_t = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ without leading to destructively large policy updates. Thus, the actor network’s loss function is expressed as

$$L(\theta) = -\mathbb{E}_t \left[\min[\lambda_t A_t^{\pi_{\text{old}}}, \text{clip}(\lambda_t, 1 - \epsilon, 1 + \epsilon) A_t^{\pi_{\text{old}}}] + \beta H[\pi_{\theta}(s_t)] \right]$$

where the operator $\mathbb{E}_t(\cdot)$ indicates a T -length empirical average over a batch of samples with $t \in [0, T - 1]$, and the entropy function $H(\cdot)$ ensures sufficient exploration, while β is a hyperparameter to reflect the relative importance of entropy. Besides, the function $\text{clip}(\cdot, 1 - \epsilon, 1 + \epsilon)$ aims to penalize over-large policy changes and clips the ratio into $[1 - \epsilon, 1 + \epsilon]$, where ϵ is a hyperparameter. Furthermore, $A_t^{\pi_{\text{old}}} = \delta_t + \gamma \delta_{t+1} + \dots + \gamma^{T-t-1} \delta_{T-1}$ is an estimator of the advantage function at timestep t , where $\delta_t = r_t + \gamma V_{\omega}(s_{t+1}) - V_{\omega}(s_t)$, and γ denotes a discount factor. Along with the update of policy π_{θ} , $V_{\omega}(s_t)$ parameterized by ω is estimated by another critic network in terms of Mean Squared Error (MSE) loss

$$L(\omega) = \mathbb{E}_t [(V_{\omega}(s_t) - V_t^{\text{targ}})^2]$$

where V_t^{targ} is the target value equals $\sum_{i=0}^{T-t-1} \gamma^i r_{t+i} + \gamma^{T-t} V_{\omega}(s_T)$. At local iteration k , the parameter update follows a standard Stochastic Gradient Descent (SGD) as

$$\theta_{k+1}^{(i)} = \theta_k^{(i)} - \eta_a \nabla L(\theta_k^{(i)}) \quad (1)$$

$$\omega_{k+1}^{(i)} = \omega_k^{(i)} - \eta_c \nabla L(\omega_k^{(i)}) \quad (2)$$

where η_a and η_c are the learning rate of the actor network and the critic network respectively.

Upon every τU local iterations (i.e., τ times of copying parameters from $\pi_{\theta_{\text{old}}}$ to π_{θ}), the communications among neighboring agents starts. Considering the possible communication bandwidth or delay restriction between agents in real-world facilities, we assume messages transmitted by agents

²Hereafter, for simplicity of representation, we omit the superscript (i) under cases where the mentioned procedure applies for any agent.

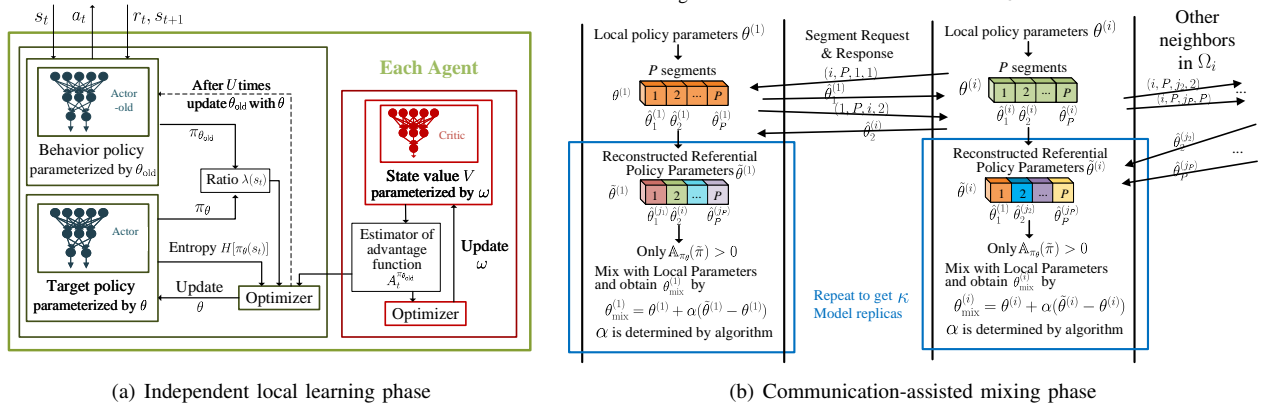


Fig. 2. The illustration of RSM-MAPPO implementation.

are limited to policy parameters θ , and agents could develop different means to derive a referential policy parameterized by $\tilde{\theta} = f(\theta^{(1)}, \dots, \theta^{(j)}, \dots)$ by exploiting the parameters from neighboring agents $\forall j \in \Omega_i$. For example, agent i could compute a referential policy $\tilde{\pi}^{(i)}$ parameterized by $\tilde{\theta}^{(i)}$ based on received parameters $\theta^{(j)}$ from $j \in \Omega_i$, and directly mix neural network parameters distributedly as

$$\theta_{\text{mix}}^{(i)} = \theta^{(i)} + \alpha(\tilde{\theta}^{(i)} - \theta^{(i)}) \quad (3)$$

where $\alpha \in [0, 1]$ is a mixture metric. Taking general parameter average mixture method in DRL-Ave [9] as an example, $\tilde{\theta}^{(i)}$ is computed as $\tilde{\theta}^{(i)} = \sum_{j \in \Omega_i} \theta^{(j)}$ and $\alpha = 1 - 1/|\Omega_i|$, which is influenced by the number of neighbors involved. Subsequently, for each agent i , $\theta^{(i)}$ and $\theta_{\text{old}}^{(i)}$ get aligned with $\theta_{\text{mix}}^{(i)}$.

B. Problem Formulation

This paper primarily targets the communication assisted mixing phase. Intuitively, an effective mixture means could better leverage the exchanged parameters to yield a superior target policy and thus benefit the learning in terms of the rewards along with the learning trajectory. In other words, the reward could be a function of the mixed policy parameters $\Theta = \{\theta_{\text{mix}}^{(1)}, \dots, \theta_{\text{mix}}^{(N)}\}$ and α . However, it remains little investigated on the feasible means to mix the exchanged parameters (or their partial segments), though it vitally affects both the communication overheads and learning performance. Therefore, by optimizing both f and α , we mainly focus on reducing the communication expenditure while maintain an acceptable cumulative rewards, that is,

$$\begin{aligned} & \min_{f, \alpha} c(v, f) \\ \text{s.t. } & \sum_t r_t(\Theta, \alpha) \geq r^{\text{thre}} \\ & \Theta \leftarrow \{\theta_{\text{mix}, k}^{(1)}, \dots, \theta_{\text{mix}, k}^{(N)}\} \\ & \theta_{\text{mix}, k}^{(i)} = \theta_k^{(i)} + \alpha(\tilde{\theta}_k^{(i)} - \theta_k^{(i)}), \quad \forall i \in \{1, \dots, N\} \\ & \tilde{\theta}_k^{(i)} = f(\theta_k^{(1)}, \dots, \theta_k^{(j)}, \dots), \quad \forall k \bmod \tau U = 0, j \in \Omega_i \end{aligned} \quad (4)$$

where r^{thre} denotes the required minimum cumulative rewards, and v indicates the size of policy parameters. Furthermore,

$c(v, f)$ denotes the communication expenditure, which is governed by the mixture function f . For example, for the whole policy parameters transmission among all agents [9], the total communication cost per round is $c(v, f) = N \times (N - 1) \times v$. Apparently, the communication cost $c(v, f)$ can be significantly reduced, if f could rely on fewer agents with reduced communication frequency. However, such a naive design possibly mitigates the positive effect of collaboration as well. Therefore, it is worthwhile to resort to a more comprehensive design of f and α to calibrate the communicating agents and content as well as regulate the mixture means, so as to provide a guarantee of performance improvement.

III. MAPPO WITH REGULATED SEGMENT MIXTURE

In this section, as shown in Fig. 2, we present the design of RSM-MAPPO, which reduces the communication overheads on the basis of not much learning performance sacrifice.

A. Algorithm Design

Consistent with the standard PPO as in Section II, agents in RSM-MAPPO undergo the same local iteration process. Meanwhile, for the communication-assisted mixing phase, RSM-MAPPO typically entails segment request & response, model replica building, and parameter mixture with theory-established performance improvement.

1) *Segment Request & Response*: Inspired by segmented pulling synchronization in DFL [14], we develop and perform a segment request & response procedure, which allows the agent to request different parts of its policy parameters from different neighbors and rebuild a mixed referential policy for aggregation. Specifically, for every communication round, each agent i breaks its policy parameters $\theta^{(i)}$ into P ($P \leq |\Omega_i|$) non-overlapping segments $\hat{\theta}_1^{(i)}, \hat{\theta}_2^{(i)}, \dots, \hat{\theta}_P^{(i)}$ as

$$\theta^{(i)} = (\hat{\theta}_1^{(i)}, \hat{\theta}_2^{(i)}, \dots, \hat{\theta}_P^{(i)}) \quad (5)$$

Notably, available segmentation strategies include, but not limited to, dividing the policy parameters according to the neural network layers, the amount of samples each agent collected, the size of total parameters, etc. Here, we consider the most intuitive parameters uniform partition to clarify this process. And for each segment $p = 1, \dots, P$, agent i randomly selects

Algorithm 1 Communication-assisted mixing phase of RSM-MAPPO Alogrithm

Input: the target policy's parameters $\theta^{(i)}$ for $i = 1, 2, \dots, N$; number of samples to estimate policy advantage M ; number of samples to evaluate FIM K ; number of replica κ ; number of segment P .

Output: $\theta_{\text{mix}}^{(i)}$ for $i = 1, 2, \dots, N$;

- 1: **Each agent** i **executes:**
- 2: **for** each replica $u = 1, 2, \dots, \kappa$ **do**
- 3: Send P pulling request (i, P, j_p, p) to nearby collaborators in Ω_i , and receive $\hat{\theta}_p^{(j_p)}$ to reconstruct $\tilde{\theta}$ as (6).
- 4: Randomly select M samples from the replay buffer of agent i under the behavior policy $\pi_{\theta_{\text{old}}}$ to estimate $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$ according to (7);
- 5: **if** $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi}) > 0$ **then**
- 6: Randomly select K samples from the replay buffer of agent i to evaluate $G(\theta^{(i)})$ according to (8).
- 7: Get the upper bound of α according to Theorem 1.
- 8: Make the mixture metric α less than the calculated upper bound, and update $\theta^{(i)}$ by (3).
- 9: **end if**
- 10: **end for**
- 11: **return** the referential policy's parameters $\theta_{\text{mix}}^{(i)}$ for $i = 1, 2, \dots, N$.

a target agent (without replacement) from its neighbors (i.e., $j_p \in \Omega_i$) to send segment request (i, P, j_p, p) , which indicates the agent i who initiates the request and its total segment number P , the target agent j_p that will receive the request and break its own policy parameters $\theta^{(j_p)}$ into also P segments, return the corresponding requested segment $\hat{\theta}_p^{(j_p)}$ in response according to the identifier p . It should be stressed that in order to reduce the complexity and facilitate the implementation, we only discuss the case as in Fig. 2 that P is the same constant for all agents and is not greater than $\max_i |\Omega_i|, \forall i$. Then, agent i could reconstruct a referential policy based on all of the fetched segments, that is,

$$\tilde{\theta}^{(i)} = (\hat{\theta}_1^{(j_1)}, \hat{\theta}_2^{(j_2)}, \dots, \hat{\theta}_P^{(j_P)}) \quad (6)$$

This step, which can be conveniently performed in parallel to make full use of the bandwidth, contributes to avoiding the model staleness, since one reconstructed model consists of different agents' latest update policy segments, thus propagating more agents' local updates through the whole system.

2) *Model Replica Building:* As it is difficult to bound the staleness of model updates, we adopt the concept of model replica into RSM-MAPPO, so as to further accelerate the propagation and ensure the model quality. Specifically, each agent i repeats the process of segment request and response for κ times, thus reconstructing κ distinctive model replicas.

3) *Parameter Mixture with Theory-Established Performance Improvement:* As the policy performance may vary significantly due to the differences in training samples of multiple agents, there might emerge some reconstructed model replicas degrading the learning performance, and a direct application of averaging mixture method in Section II possibly makes the aforementioned procedures in vain. Instead, based on our previous works [9], we derive the following mixture

metric to justify the effectiveness of a model replica and only select the contributive ones. Beforehand, we give the following useful theorem [9].

Theorem 1: For a PPO agent with a current target policy π_{θ} and a referential policy $\tilde{\pi}$ parameterized by θ and $\tilde{\theta}$ respectively, if

- 1) $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi}) > 0$
- 2) $0 < \alpha < \left[2 \left(\frac{\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})}{C} \right)^{\frac{1}{2}} / \left[(\tilde{\theta} - \theta)^T G(\theta) (\tilde{\theta} - \theta) \right]^{\frac{1}{2}} \right]$

the cumulative rewards are guaranteed to be improved through updating θ to $\tilde{\theta}$ according to (3). Notably, $C = \frac{2\varepsilon\gamma}{(1-\gamma)^2}$, $\varepsilon = \max_{s_t} \max_{a_t} |\delta_t|$. $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$ is defined as the expectation of the advantage function along the $\tilde{\pi}$ -yielded learning trajectory, and can be approximated as the expectation of the multiplication of policy gain $\tilde{\pi} - \pi_{\theta}$ and 1-step advantage function δ_t along with the $\pi_{\theta_{\text{old}}}$ -yielded learning trajectory. Meanwhile, $G(\theta)$ is the Fisher Information Matrix (FIM) of policy parameters θ .

Based on Theorem 1, we can verify the contribution of a model replica by computing $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$, and get the upper bound of α by further computing $G(\theta)$ from Monte-Carlo simulations, that is,

$$\mathbb{A}_{\pi_{\theta}}(\tilde{\pi}) \approx \mathbb{E}_t \left[\frac{\tilde{\pi}(a_t|s_t) - \pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \right] \delta_t \quad (7)$$

$$G(\theta) \approx \mathbb{E}_t \left[\left(\frac{\partial \log \pi_{\theta}(a_t|s_t)}{\partial \theta} \right) \left(\frac{\partial \log \pi_{\theta}(a_t|s_t)}{\partial \theta} \right)^{\top} \right] \quad (8)$$

Afterwards, we can select and mix those model replicas with positive $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$, which means the agent can benefit from mixing its policy parameters π_{θ} with the reconstructed referential policy $\tilde{\pi}$. More aggressively, it is also feasible to merge the model replica with the largest $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$ only. Besides, since $G(\theta)$ is a positive definite matrix, the mixture metric α will enlarge with the increase of $\mathbb{A}_{\pi_{\theta}}(\tilde{\pi})$. Thus, a better referential policy contributes to faster learning as well.

Finally, we summarize the details of RSM-MAPPO in Algorithm 1.

B. Discussions of Communication overheads

In regards to the communication overheads per segment request, RSM-MAPPO costs v/P amount of data transmission via D2D communications. Therefore, the total amount of communications overheads per round equals $N \times v$, which is $N-1$ times less than that in [9]. Meanwhile, by simultaneously requesting P agents, it benefits the sufficient use of the bandwidth and enhances the capability to overcome possible channel degradation. On the other hand, for cases with κ model replicas, the communication overheads per round turns to $N \times \kappa \times v$, which is $\frac{N-1}{\kappa}$ times less than that in [9], but improves the learning performance.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Experimental Settings

In this part, we consider the simulation scenario "Figure Eight", a widely-used mixed-autonomy traffic control scenario, to testify the performance (i.e., maximizing the cumulative

TABLE II
SYSTEM PARAMETERS.

Parameters	Symbol	Value
Total time-steps of an epoch	E	1500
Number of timesteps for a mini-batch	T	250
Number of PPO iterations in a mini-batch	U	3
Number of samples to evaluate FIM	K	50
Number of samples to estimate $\mathbb{A}_{\pi_\theta}(\tilde{\pi})$	M	200
Learning rate of actor network	η_a	$2.5 \cdot 10^{-5}$
Learning rate of critic network	η_c	$5 \cdot 10^{-5}$
Discount factor	γ	0.9
Entropy coefficient	β	0.01
Coefficient of communication internals	τ	1
Number of segments	P	4
Number of model replicas	κ	2

rewards) of DRL [15]. There are totally 14 vehicles running circularly along a one-way lane that resembles the shape of figure “8”. These include 5 emulated human-driving vehicles, controlled by Simulation of Urban Mobility (SUMO) with a microscopic car-following model named Intelligent Driver Model (IDM) [16], and 9 IRL-controlled CAVs, which simultaneously maintain dedicated links to update their parameters through the D2D channel. Besides, the scenario is modified to assign the limited partial-observation of global environment as the state of each vehicle, including the position and speed of its own, the vehicle ahead and behind. Meanwhile, each CAV’s action is a continuous variable representing the speed acceleration or deceleration normalized between $[-1, 1]$. In order to reduce the occurrence of collisions and promote the traffic flow to the maximum desired speed, the reward function is $\mathcal{R} = \frac{\max\{\|\mathbf{v}_{de}\| - \|\mathbf{v}_{ac} - \mathbf{v}_{de}\|, 0\}}{\|\mathbf{v}_{de}\|}^3$, where $\mathbf{v}_{de} \in \mathbb{R}^{14}$ and $\mathbf{v}_{ac} \in \mathbb{R}^{14}$ represent the desired velocity and actual velocity of all vehicles in the system respectively. In addition, the current epoch will be terminated once a collision occurs. We perform tests every 10 epochs and take the average of accumulated rewards in a testing epoch as average reward. Besides, all results are produced using the average of 5 repetitions. The main parameters used in simulations are listed in Table II.

B. Evaluation Metrics

Besides average reward, we also adopt other metrics to extensively evaluate communication efficiency of RSM-MAPPO.

- We use ρ_{total} to represent the total number of reconstructed referential policy $\tilde{\pi}$ (i.e., all model replicas) until convergence, that is, the inflexion point of average reward curve. Moreover, we use ρ_{ef} to indicate the number of effectively reconstructed referential policy (i.e., contributive model replicas selected to mix). Correspondingly, we further define the ratio $\rho_r = \rho_{ef}/\rho_{total}$ to reflect the utilization rate of reconstructed policies.
- We use ψ to indicate the communication overheads (in terms of v) until convergence. Mathematically, as the number of communication rounds until convergence can

³Notably, we assume complete knowledge of individual vehicle speeds at each vehicle here. Beyond the scope of this paper, some value-decomposition method like [17] can be further leveraged to derive a decomposed reward, so as to loosen such a strict requirement.

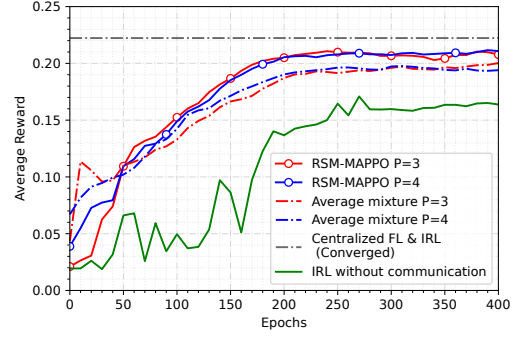


Fig. 3. Average reward under different methods.

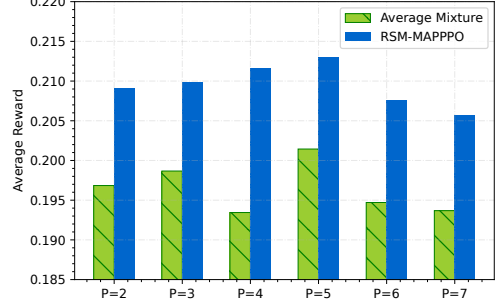


Fig. 4. Average reward of average mixture & RSM-MAPPO with different P .

be computed as $C_0 = \rho_{total}/(N \times \kappa)$, the communication overheads equal $\psi = C_0 \times (N \times \kappa \times v) = \rho_{total} \times v$.

Intuitively, the average reward and communication efficiency will be determined by the function design of f , as well as the number of model segments and the model replica (i.e., P and κ). Besides, the communication overheads are also affected by the coefficient of communication intervals τ .

C. Simulation Results

Fig. 3 first examines the average reward with RSM-MAPPO $\kappa = 2$. It can be observed from Fig. 3, the curve of IRL without communications implies that simply extending IRL to multi-agent scenarios without any cooperation cannot solve complex tasks. As a comparison, other methods adding communications among agents can clearly boost the learning performance in terms of training efficiency and stability. Besides, we use the well trained model under combination of centralized FL and IRL [5] as the optimal baseline. It can be seen that our RSM-MAPPO which is performed in a fully distributed training process, approaches the converged performance of the centralized method. Meanwhile, compared with simply average mixture, which directly takes the average of all replicas, RSM-MAPPO also yields superior converged average reward, which can also be further validated in Fig. 4. On the other hand, Fig. 3 shows that partitioning the model into different segments (i.e., different P) leads to similar convergence trend. However, the converged average reward is relevant to the exact value of P , as demonstrated in Fig. 4, which investigates this influence. Specifically, the final converged average reward becomes higher at first with the increase of P from 2 to 5, but then decreases when $P = 6$ and $P = 7$. The performance degradation in the latter cases is because that the aggregation

TABLE III
AVERAGE REWARD & COMMUNICATION EFFICIENCY OF RSM-MAPPO WITH RESPECT TO THE METRICS IN SECTION IV-B.

Method	τ	P	κ	Average Reward	ρ_{total}	ψ	ρ_{ef}	ρ_r
Average mixture			3	0.1987	$2.4948 \cdot 10^4$	$2.4948 \cdot 10^4 \times v$	$2.4948 \cdot 10^4$	100%
			4	0.1934	$2.7108 \cdot 10^4$	$2.7108 \cdot 10^4 \times v$	$2.7108 \cdot 10^4$	100%
RSM-MAPPO		1	1	0.2121	$1.3010 \cdot 10^4$	$1.3010 \cdot 10^4 \times v$	$1.3010 \cdot 10^3$	40.402%
			2	0.2116	$2.2788 \cdot 10^4$	$2.2788 \cdot 10^4 \times v$	$9.110 \cdot 10^3$	39.977%
			4	0.2137	$5.4216 \cdot 10^4$	$5.4216 \cdot 10^4 \times v$	$2.0127 \cdot 10^4$	37.124%
			8	0.2128	$8.2512 \cdot 10^4$	$8.2512 \cdot 10^4 \times v$	$3.0798 \cdot 10^4$	37.325%
		3	2	0.2110	$6.498 \cdot 10^3$	$1.300 \cdot 10^4 \times v$	$2.711 \cdot 10^4$	41.721%
			4	0.2080	$3.240 \cdot 10^3$	$6.480 \cdot 10^3 \times v$	$1.333 \cdot 10^3$	41.141%
			8	0.2100	$2.376 \cdot 10^3$	$4.752 \cdot 10^3 \times v$	$1.016 \cdot 10^3$	42.761%
			15					

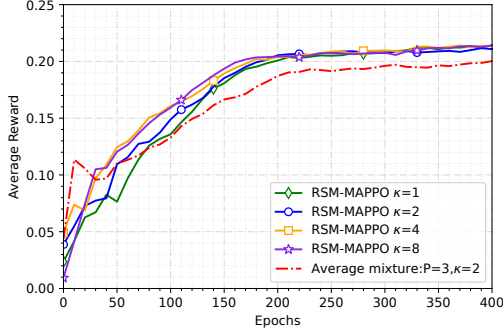


Fig. 5. Average reward of RSM-MAPPO with different κ .

target of reconstructed policy parameters for an over-large P is mottled and loses integrality.

In addition, Fig. 5 studies the impact of the number of model replicas κ on average reward. As shown in Fig. 5, an increase of κ could accelerate the convergence of training process, without apparent influences on the converged average reward. On the other hand, the improvement in the convergence rate comes at the cost of increased communication overheads ψ , which is listed in Table III. With the increase of κ , both ψ and ρ_{ef} increase, but the ratio ρ_r does not increase scalely. Therefore, the trade-off between convergence rate improvement and communication overheads need to be further considered. Furthermore, we testify the performance under different values of communication interval τ . Due to the space limitation, the results along with the detailed comparison of the communication efficiency is summarized in Table III. The communication overheads ψ are reduced by τ times compared with $\rho_{\text{total}} \times v$, resulting into higher ρ_r for a larger τ .

V. CONCLUSIONS

In this paper, we have proposed a communication-efficient algorithm RSM-MAPPO to deal with the excessive communication overheads among distributed MARL. By delving into the policy parameter mixture function, RSM-MAPPO has provided a novel means to leverage and boost the effectiveness of distributed multi-agent collaboration. In particular, RSM-MAPPO has successfully transformed the classical means of complete parameter exchange into segment-based request and response, which significantly facilitates the construction of multiple model replicas and simultaneously captures enhanced learning diversity. Moreover, in order to avoid performance-harmful parameter mixture, RSM-MAPPO has leveraged a

theory-established regulated mixture metric to select the contributive replicas with positive relative policy advantage only. Finally, extensive simulations have demonstrated the effectiveness of this design. In the future, we will extend this regulated segment mixture paradigm to more RL algorithms to verify its generalization.

REFERENCES

- [1] B. R. Kiran, *et al.*, “Deep Reinforcement Learning for Autonomous Driving: A Survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.
- [2] A. R. Kreidieh, *et al.*, “Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning,” in *Proc. ITSC*, Maui, HI, USA, Nov. 2018.
- [3] T. Shi, *et al.*, “Efficient connected and automated driving system with multi-agent graph reinforcement learning,” *arXiv preprint arXiv:2007.02794*, 2020.
- [4] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in *Proc. Int. Conf. Mach. Learn.*, University of Massachusetts, Amherst, Jun. 1993.
- [5] X. Xu, *et al.*, “The gradient convergence bound of federated multi-agent reinforcement learning with efficient communication,” *arXiv preprint arXiv:2103.13026*, 2021.
- [6] J. Foerster, *et al.*, “Learning to communicate with deep multi-agent reinforcement learning,” in *Proc. NeurIPS*, Barcelona, Spain, Dec. 2016.
- [7] J. Jiang, *et al.*, “Learning attentional communication for multi-agent cooperation,” in *Proc. NeurIPS*, Montréal, Canada, Dec. 2018.
- [8] W. Kim, *et al.*, “Communication in multi-agent reinforcement learning: Intention sharing,” in *Int. Conf. Learn. Represent.*, Virtual, Online, May 2021.
- [9] X. Xu, *et al.*, “Trustable Policy Collaboration Scheme for Multi-Agent Stigmergic Reinforcement Learning,” *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 823–827, Apr. 2022.
- [10] L. Barbieri, *et al.*, “Communication-efficient Distributed Learning in V2X Networks: Parameter Selection and Quantization,” in *Proc. IEEE Globecom*, Rio de Janeiro, Brazil, Dec. 2022.
- [11] C. Hu, *et al.*, “Decentralized federated learning: A segmented gossip approach,” *arXiv preprint arXiv:1908.07782*, 2019.
- [12] P. Kairouz, *et al.*, “Advances and open problems in federated learning,” *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, Jun. 2021.
- [13] J. Schulman, *et al.*, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [14] I. Hegedűs, *et al.*, “Gossip learning as a decentralized alternative to federated learning,” in *19th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, Kongens Lyngby, Denmark, Jun. 2019.
- [15] E. Vinitzky, *et al.*, “Benchmarks for reinforcement learning in mixed-autonomy traffic,” in *Proc. CoRL*, Zürich, Switzerland, Oct. 2018.
- [16] M. Treiber, *et al.*, “Congested traffic states in empirical observations and microscopic simulations,” *Physical review E*, vol. 62, no. 2, p. 1805, Aug. 2000.
- [17] B. Xiao, *et al.*, “Stochastic graph neural network-based value decomposition for multi-agent reinforcement learning in urban traffic control,” in *Proc. IEEE VTC 2023-Spring*, Florence, Italy, Jun. 2023.