Communication-Efficient Soft Actor–Critic Policy Collaboration via Regulated Segment Mixture

Xiaoxue Yu, *Student Member, IEEE*, Rongpeng Li, *Senior Member, IEEE*, Chengchao Liang, and Zhifeng Zhao, *Member, IEEE*

***Abstract*—Multiagent reinforcement learning (MARL) has emerged as a foundational approach for addressing diverse, intelligent control tasks in various scenarios like the Internet of Vehicles, Internet of Things, and unmanned aerial vehicles. However, the widely assumed existence of a central node for centralized, federated learning-assisted MARL might be impractical in highly dynamic environments. This can lead to excessive communication overhead, potentially overwhelming the system. To address these challenges, we design a novel communication-efficient, fully distributed algorithm for collaborative MARL under the frameworks of soft actor–critic (SAC) and decentralized federated learning (DFL), named regulated segment mixture-based multiagent SAC (RSM-MASAC). In particular, RSM-MASAC enhances multiagent collaboration and prioritizes higher communication efficiency in dynamic systems by incorporating the concept of segmented aggregation in DFL and augmenting multiple model replicas from received neighboring policy segments, which are subsequently employed as reconstructed referential policies for mixing. Distinctively diverging from traditional reinforcement learning (RL) approaches, RSM-MASAC introduces new bounds under the framework of maximum entropy reinforcement learning (MERL). Correspondingly, it adopts a theory-guided mixture metric to regulate the selection of contributive referential policies, thus guaranteeing soft policy improvement during the communication-assisted mixing phase. Finally, the extensive simulations in mixed-autonomy traffic control scenarios verify the effectiveness and superiority of our algorithm.**

***Index Terms*—Communication-efficient, multiagent reinforcement learning (MARL), regulated segment mixture (RSM), soft actor–critic (SAC).**

I. Introduction

RECENTLY, deep reinforcement learning (DRL) has gained significant traction in addressing complex, real-world applications, contingent on formulated Markov decision processes (MDPs) [2], [3]. Naturally, multiple collaborative DRL agents can form a scalable multiagent reinforcement learning (MARL)-empowered system, efficiently coordinating sequential decision-making for complex scenarios [4], [5], [6], [7], [8]. Notable examples include fleet management and traffic control in the Internet of Vehicles (IoV), unmanned aerial vehicles (UAVs), and multirobot systems, as well as distributed resource allocation in the Internet of Things (IoT).

A. Problem Statement and Motivation

In general, most MARL works adopt a centralized training and decentralized execution (CTDE) architecture [9], [10], [11], [12]. Federated reinforcement learning (FRL) [13], [14], [15], [16] incorporates the continuous learning process in DRL with periodic model updates (i.e., exchanging gradients or parameters) from federated learning (FL), thus more competently coping with generalization difficulties during deployment [17], [18], such as variations in both physical and social environments. By keeping sensitive information localized, FRL enables instantaneous decision-making and promotes collaborative learning among independent agents. However, in most highly dynamic scenarios with high mobility, intermittent connectivity, and decentralized nature [19], [20], the common assumption of a super-centralized training controller in centralized FL (CFL) becomes impractical, threatening both the stability and timeliness of overall learning performance [21]. Besides, generalization difficulties [18], [22], such as variations in external environments, necessitate continuous training, or fine-tuning of decision-making models for specific environments or tasks with minimal online data requirements during deployment.

In this way, there is growing research on peer-to-peer architecture's decentralized FL (DFL) [23], focusing on real-time decentralized training/fine-tuning of deep neural network (DNN) models among DRL agents [24], [25], [26]. This can be viewed as a distributed stochastic gradient descent (SGD) optimization problem, in which the aggregation of exchanged DNN parameters acquired from the periodic communication phase typically uses a simplistic parameter average mixture approach. In FRL, these exchanged and averaged parameters pertain to the DRL agent's policy network, which approximates the policy distribution through a parameterized DNN. In addition, the parameter exchange can be implemented via device-to-device (D2D) or Vehicle-to-Vehicle (V2V) collaboration channels within communication range, common in many MARL works [7], [17], [27], [28]. However, the frequent information exchanges generate substantial communication overhead as the number of agents increases. Some DFL

Received 9 July 2024; revised 8 August 2024 and 25 September 2024; accepted 10 October 2024. Date of publication 16 October 2024; date of current version 6 February 2025. This article was presented in part at IEEE Globecom 2023 [DOI: 10.1109/GLOBECOM54140.2023.10437274]. *(Corresponding author: Rongpeng Li.)*

Xiaoxue Yu and Rongpeng Li are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310013, China (e-mail: sdwhyxx@zju.edu.cn; lirongpeng@zju.edu.cn).

Chengchao Liang is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: liangcc@cqupt.edu.cn).

Zhifeng Zhao is with Zhejiang Lab, Hangzhou 311121, China, and also with the the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310013, China (e-mail: zhaozf@zhejianglab.com).

Digital Object Identifier 10.1109/JIOT.2024.3481257

variants improve communication efficiency by reducing the number of communication rounds [29], [30], [31] or the communication workload per round [25], [32], [33], [34], but this can intensify the variability of local model updates, potentially leading to an inferior aggregated model post simple model parameter averaging [35]. This issue is more pronounced in online DRL frameworks since DRL agents interact more frequently with the environment compared to offline supervised learning. Incremental data can magnify learning discrepancies among agents and reduce fault tolerance. Interestingly, few studies have focused on the policy performance improvement issue of DRL with its combination with FL. Despite the simplicity and straightforward aggregation of mixed policies in parameter averaging, not all communicated packets in MARL contribute effectively. During the communication-assisted mixing phase, MARL awaits a revolutionary policy parameter mixture method and corresponding metrics to regulate the aggregation of exchanged model updates, ensuring robust policy improvement for security and safety.

On the other hand, there is a contradiction between limited and costly sample availability and the need for fast learning speeds, that is, MARL algorithms shall fully explore and adapt to dynamic environments while minimizing the demand for extensive and costly online learning data. This aligns with the principles of maximum entropy reinforcement learning (MERL), such as soft Q-learning [36] and soft actor–critic (SAC) [37], [38], which transforms the traditional reward maximum into both expected return and the expected entropy of the policy. The incorporated entropy in optimization redefines value function and policy optimization objective from the ground up. Albeit its enhancement to sample efficiency, adaptability, and robustness in dynamic and uncertain environments, it overturns the proof of the traditional performance improvement bound [39], [40], [41], [42]. Among MERL algorithms, SAC [37], [38] is particularly well-suited for highly dynamic and uncertain environments due to multifolded reasons. First, it leads to higher sample efficiency through off-policy learning. Second, it enables automatic entropy adjustment via optimization of the temperature parameter. Finally, by effectively combining value function learning with policy optimization, it significantly enhances stabilization and adaptability to continuous, complex, high-dimensional action space.

Therefore, this work is dedicated to reanalyzing efficient communication within the MERL and DFL framework. Prominently, a practical policy mixture method, which is underpinned by a rigorously derived mixture metric, is devised to enable SAC to achieve smooth and reliable parameter updates, not only during independent local learning phase but also throughout the communication-assisted mixing phase.

B. Contribution

In this article, we propose the regulated segment mixture-based multiagent SAC (RSM-MASAC) algorithm, tailored for training DRL agents under highly dynamic scenarios while addressing communication overhead challenges inherent in DFL. Our primary contributions include the following.

1) For the highly dynamic setting, the proposed RSM-MASAC algorithm effectively combines communication-efficient DFL with MERL. The algorithm enables agents to receive segments of policy networks' parameters from neighbors within their communication range. These segments are used to constitute referential policies for strategically designed selective parameter mixture, ensuring credible performance improvement during the communication-assisted mixing phase while maintaining sufficient exploration in the local learning phase.
2) In terms of mixing a current policy and a referential policy, we derive a new, more generalized mixed policy improvement bound, which successfully tackles the analysis difficulties arising from the incorporation of redefined soft value function, dual policy optimization objective, and the logarithmic term of the policy for entropy maximum in MERL. Therefore, our work sets the stage for theoretically evaluating the performance of the mixed policy under distributed SGD optimization.
3) Instead of a simple parameter average in DFL, we bridge the relationship between DNN parameter gradient descent and MERL policy improvement, and regulate the selection of contributive referential policies by deriving a manageable, theory-guided mixture metric. Hence, it enhances the stability and practicality of directly mixing policy parameters during the communication-assisted mixing phase.
4) Through extensive simulations in the traffic speed control task, a typical MARL-based IoV scenario, our proposed RSM-MASAC algorithm could approach the converged performance of centralized FMARL [14] in a distributed manner, outperforming parameter average methods as in DFL [25], [34], thus confirming its effectiveness.

C. Related Works

In multiagent systems (MASs), independent reinforcement learning (IRL) [48], [49], [50], which relies solely on agents' local perceptions without any collaboration, has been extensively studied and often used as a baseline due to its varied policy performance, unstable learning and uncertain convergence [14]. Examples of IRL methods include IQL [49], IAC [11], IA2C [51], independent PPO (IPPO) [52], etc.

As an extension of IRL, distributed cooperative reinforcement learning (RL) enhances agents' collective capabilities and efficiency by collaboratively seeking near-optimal solutions through limited information exchange with others, as illustrated in Fig. 1. Notably, the exchanged contexts can be rather different and possibly include approximated value functions in [43] and [44], rewards or even maximal Q-values on each state-action pair in [45]. Besides, given the clear evidence [53], [54] that experiences from homogeneous, independent learning agents in MAS can contribute to efficiently learning a commonly shared DNN model, the direct exchange of model updates during FL communication phase [13], [14], [55] is a viable way to indirectly integrate

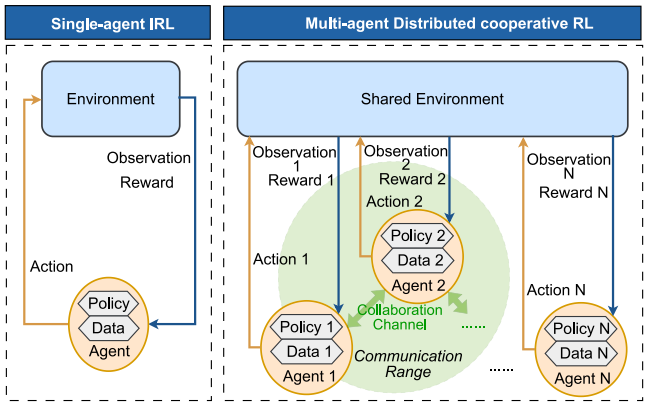


Fig. 1. Illustration of single-agent IRL and multiagent distributed cooperative RL.

information and enhance cooperation among IRL. In addition to the commonly used centralized architecture that is less suitable for actual dynamic environments, the peer-to-peer DFL paradigm, where clients exchange their local model updates only with their neighbors to achieve model consensus, emerges as an appealing alternative. This approach can be viewed as distributed SGD optimization, where the significant communication expenditure cannot be overlooked. In that regard, some researchers have developed strategies to reduce communication frequency by aggregating more local updates before one round communication [29] or multiround communication [30], [31]. Besides, reducing the number of parameters transmitted from local models or implementing selective model synchronization is also tractable. For instance, Watcharapichat et al. [32] divided local gradients into several disjoint partitions, with only a subset being exchanged in any given communication round. Barbieri et al. [25] puts forward a randomized selection scheme for forwarding subsets of local model parameters to their one-hop neighbors. Furthermore, Barbieri et al. [33] suggested propagating top-k layers with higher normalized squared gradients, which may convey more information about the local data to neighbors. Meanwhile, Hu et al. [34] introduced a segmented gossip approach that involves synchronizing only model segments, thereby substantially splitting the communication expenditure.

Moreover, while the above distributed SGD optimization methods and most works in actual IoV, IoT, and UAV applications [4], [19], [20], [24], [56] assume ideal communication between cooperative agents, achieving exact and perfect information sharing without compromising privacy concerns may not be feasible with suboptimal wireless transmission links. Beyond complex, customized, and costly cryptographic schemes [35], [57], most research [46], [47], [58] suggests local quantization of data before transmission, effectively shielding raw data from exposure. Additionally, quantization, along with compression or sparsification [31], [59], [60], can significantly reduce the message size, with corresponding convergence analyses detailed in [30], [46], and [47].

Notably, when it comes to the method of mixing DNN parameters, these aforementioned DFL works generally adopt a simplistic averaging approach, which is familiar in parallel distributed SGD methods. However, when such an approach is directly applied to FRL, the crucial relationship between parameter gradient descent and policy improvement will be overshadowed [39]. Consequently, the corresponding mixture lacks proper, solid assessment means to prevent potential harm to policy performance [15]. In other words, directly using this kind of naive combination of communication efficient DFL and RL to enhance individual policy performance in IRL appears inefficient.

Regarding this issue, we can draw inspiration from the conservative policy iteration algorithm [40], which leverages the concept of policy advantage as a crucial indicator to gauge the cumulative reward improvement and applies a direct mixture update rule for policy distributions in pursuit of an approximately optimal policy. Moreover, the mixture metric utilized in the update rule is also investigated to prevent overly aggressive updates toward risky directions, as excessively large policy updates often lead to significant policy performance deterioration [15]. TRPO [42] substitutes the mixture metric with Kullback–Leibler (KL) divergence, a measure that quantifies the disparity between current and updated policy distributions, facilitating the learning of monotonically improving policies. Kuba et al. [41] extended this work into cooperative MARL settings. However, directly mixing policy distributions is often an intractable endeavor. To implement this procedure in practical settings with parameterized policies in DRL, Xu et al. [39] further simplified the KL divergence to the parameter space through fisher information matrix (FIM), so as to improve the policy performance by directly mixing DNN parameters. It focuses on stable policy updates throughout the communication-assisted mixing phase, but its analysis is confined to the traditional policy iteration-based RL algorithms, which aim to maximize the expected return only.

Transitioning to the realm of MERL, the integration of the entropy maximization marks a significant paradigm shift, fundamentally redefining the criteria for policy improvement due to its dual objective of optimizing cumulative rewards and maintaining a high level of exploration. However, traditional RL methodologies, particularly those based on PPO, are insufficient to address the new dynamics interplayed between reward maximization and exploratory behavior imposed by the entropy maximization in MERL, as demonstrated in many works [37], [38], [61], [62]. On the other hand, it is also imperative to implement an appropriate and manageable mixture metric with monotonic policy improvement property to maximize the practicality of directly mixing policy parameters during the communication-assisted mixing phase. Such a metric must guarantee the efficacy of the resultant mixed policy in terms of policy improvement. Crucially, the expected benefits of the mixed policy must be assessed before proceeding with actual policy mixing. Specifically, the mixed policy improvement should be evaluated against the referential policy, whose parameters are received from neighbors in DFL. Only after the anticipated benefits are confirmed should the operation to evaluate the established metric for mixing DNN parameters commence. Moreover, we have also summarized the key differences between our algorithm and relevant literature in Table I. In conclusion, it is vital to reformulate the

TABLE I
SUMMARY OF DIFFERENCES WITH RELATED LITERATURE

References	Maximum Entropy	Policy Improvement Guarantee	Collaboration via Communication	Efficient Communication	Brief Description
[36]–[38]	●	○	○	○	Only single-agent RL algorithm
[40], [42]	○	●	○	○	
[43]–[45]	○	○	●	○	Over frequent and complex communication
[25], [29]–[34] [46], [47]	○	○	●	●	Oversimplified parameter mixture method
[39], [41]	○	●	●	○	Only under traditional policy iteration-based approximately optimal RL's performance guarantee analysis
[1]	○	●	●	●	
This work	●	●	●	●	Combining communication efficient DFL into MARL collaboration under maximum entropy framework with theory-established regulated mixture metrics and performance improvement bound

Notations: ○ *indicates not included;* ● *indicates fully included.*

theoretical analysis within the combination of DFL and MERL framework. More particularly, given its superiority among MERL algorithms, SAC [37], [38] is chosen as a showcase offering robust theoretical underpinnings for practical IoV, IoT, and UAV applications.

D. Paper Organization

The remainder of this article is organized as follows. In Section II, we present preliminaries of SAC algorithm and main notations used in this article. Then, we introduce the system model and formulate the problem in Section III. Afterward, we provide the mixed performance improvement bound theorem of FRL communication under MERL in Section IV, and elaborate on the details of the proposed RSM-MASAC algorithm in Section V. In Section VI, we present the simulation settings and discuss the experimental results. Finally, Section VII concludes this article and discusses future work.

II. PRELIMINARY

Beforehand, we summarize the mainly used notations in Table II.

We consider the standard RL setting, where the learning process of each agent can be formulated as an MDP. During the interaction with the environment, at each time step t, an RL agent observes a local state s_t from local state space $\mathcal{S}$, and chooses an action a_t from individual action space $\mathcal{A}$ according to the policy $\pi(\cdot|s_t) \in \Pi$, which specifies a conditional distribution of all possible actions given the current state s_t, and Π is the policy space. Then the agent receives an individual reward r_t, calculated by a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [0, R]$, and the environment transforms to a next state $s_{t+1} \sim p(\cdot|s_t, a_t)$. A trajectory starting from s_t is denoted as $\tau_t = (s_t, a_t, s_{t+1}, a_{t+1}, \ldots)$. Besides, considering an infinite-horizon discounted MDP, the visitation probability of a certain state s under the policy π can be summarized as $d_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s; \pi)$, where $P(s_t = s; \pi)$ is the visitation probability of the state s at time t under policy π.

Different from traditional RL algorithms aiming to maximize the discounted expected total rewards only, SAC [37], [38] additionally seeks to enhance the expected policy entropy. To be specific, with the redefined soft state value function $V^\pi(s_0) \coloneqq \mathbb{E}_{\tau_0 \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha H(\pi(\cdot|s_t)))\right]$ on top of the policy entropy $H(\pi(\cdot|s)) = -\mathbb{E}_{a\sim\pi} \log \pi(a|s)$, the task objective of SAC can be formulated as

$$\max \eta(\pi) = \mathop{\mathbb{E}}_{s_0 \sim \rho_0}\left[V^\pi(s_0)\right] \tag{1}$$

where ρ_0 is the distribution of initial state s_0 and $\alpha \in (0, \infty)$ is a temperature parameter determining the relative importance of the entropy term versus the reward. Obviously, when $\alpha \to 0$, SAC gradually approaches the traditional RL. Meanwhile, the soft state-action value can be expressed as $Q^\pi(s_t, a_t) \coloneqq r_t + \mathbb{E}_{\tau_{t+1}\sim\pi}\left[\sum_{l=t+1}^{\infty} \gamma^{l-t}(r_l + \alpha H(\pi(\cdot|s_l)))\right]$ [37], [38], which does not include the policy entropy of current time step but includes the sum of all future policy entropy and the sum of all current and future rewards. Consistently, the state-action advantage value under policy π is

$$A_\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t). \tag{2}$$

Accordingly, SAC maximizes (1) based on soft policy iteration, which alternates between soft policy evaluation and soft policy improvement.

TABLE II
MAJOR NOTATIONS USED IN THIS ARTICLE

Notation	Definition
$s_t^{(i)}, a_t^{(i)}, r_t^{(i)}$	Local state, individual action and reward of agent i at time step t.
$\pi, \tilde{\pi}, \pi_{\mathbf{mix}}$	Current target policy distribution, referential target policy distribution and the mixed policy distribution, which represents the probabilities of selecting each possible action given a state.
$\theta, \tilde{\theta}, \theta_{\mathbf{mix}}$	The parameter of specific neural network that parameterizes the target policy distribution, referential target policy distribution and the mixed policy distribution, respectively.
p, P	Index of segments and segmentation granularity, $p = 1, 2, \cdots, P$.
Ω_i	Set of one-hop neighbors within the communication range of agent i.
ζ	Mixture metric of current policy parameter vector and referential policy parameter vector.
α	Temperature parameter of policy entropy.
ϱ	Target smoothing coefficient of target Q networks.
κ	Predefined model replica number.
U	Communication interval determined by specified iterations of the local policy.
υ	Transmission bits of the policy parameters.
ψ	Communication consumption.

1) For given π, soft policy evaluation implies that Q^{π} is learned by repeatedly applying soft Bellman operator $\mathcal{T}^{\pi}$ to the real-valued estimate Q, given by

$$\mathcal{T}^{\pi}Q(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1}\sim p(\cdot|s_t,a_t)}\big[V(s_{t+1})\big] \tag{3}$$

where

$$V(s_t) = \mathbb{E}_{a_t\sim\pi}[Q(s_t, a_t)] + \alpha H(\pi(\cdot|s_t)). \tag{4}$$

With $Q^{k+1} = \mathcal{T}^{\pi}Q^k$, as $k \to \infty$, Q^k will converge to the soft Q function Q^{π} of π, as proven in [38].

2) In the soft policy improvement step, the goal is to find a policy π_{new} superior to the current policy π_{old}, in terms of maximizing (1). Specifically, for each state, SAC updates the policy as

$$\begin{aligned}\pi_{\text{new}} &= \arg\min_{\pi\in\Pi} D_{\text{KL}}\left(\pi(\cdot|s_t)\|\frac{\exp\left(\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s_t,\cdot)\right)}{Z^{\pi_{\text{old}}}(s_t)}\right)\\ &= \arg\max_{\pi\in\Pi}\mathbb{E}_{a_t\sim\pi}\left[Q^{\pi_{\text{old}}}(s_t, a_t) - \alpha\log\pi(a_t|s_t)\right]\end{aligned} \tag{5}$$

where $D_{\text{KL}}(\cdot)$ denotes the KL divergence, while the partition function $Z^{\pi_{\text{old}}}$ normalizes the distribution. The last equality in (5) is due to that $Z^{\pi_{\text{old}}}$ has no contribution to gradient with respect to the new policy, it can thus be ignored. As unveiled in Appendix-A, through the update rule of (5), $Q^{\pi_{\text{new}}}(s_t, a_t) \geq Q^{\pi_{\text{old}}}(s_t, a_t)$ is guaranteed for all $(s_t, a_t) \in \mathcal{S}\times\mathcal{A}$. Notably, the proof of soft policy improvement detailed in Appendix-A additionally serves as a confirmation for policy improvement of regulated segment mixture (RSM), which will be elaborated upon later.

Finally, with repeated application of soft policy evaluation and soft policy improvement, any policy $\pi \in \Pi$ will converge to the optimal policy π^* such that $Q^{\pi^*}(s_t, a_t) \geq Q^{\pi}(s_t, a_t)$, $\forall(s_t, a_t) \in \mathcal{S}\times\mathcal{A}$, and the proof can be found in [37] and [38].

III. System Model and Problem Formulation

A. System Model

We primarily consider a system consisting of N agents empowered by the MASAC learning, which encompasses an independent local learning phase and a communication-assisted mixing phase. In the first phase, we use SAC algorithm for each IRL agent i, $i \in \{1, 2, \ldots, N\}$. That is, agent i senses partial status $s_t^{(i)}$ and has its local policy $\pi^{(i)}$ approximated by neural networks, and parameterized by $\theta \in \mathbb{R}^d$. It collects samples $\langle s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)}\rangle$ in replay buffer $\mathcal{D}^{(i)}$, and randomly samples a mini-batch $\Phi^{(i)}$ for local independent model updates.[1] Subsequently, in the second phase, each agent i interacts with its one-hop neighbors $j \in \Omega_i$ within communication range, so as to reduce the behavioral localities of IRL and improve their cooperation efficiency.

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

1) Local Learning Phase: Algorithmically, in the local learning phase with SAC, parameterized DNNs are used as approximators for policy and soft Q-function. Concretely, we alternate optimizing one network of policy π parameterized by θ and two soft Q networks parameterized by ω_1 and ω_2, respectively. Besides, there are also two target soft Q networks parameterized by $\bar{\omega}_1$ and $\bar{\omega}_2$, obtained as an exponentially moving average of current Q network weights ω_1, ω_2. Yet, only the minimum Q value of the two soft Q-functions is used for the SGD and policy gradient. This setting of two soft Q-functions will speed up training while the use of target Q can stabilize the learning [37], [38], [62]. For training θ and $\omega \in \omega_1, \omega_2$, agent randomly samples a batch of transition tuples from the replay buffer $\mathcal{D}$ and performs SGD. The parameters of each soft Q network ω_x, $\forall x = 1, 2$ are updated through minimizing the soft Bellman residual error, that is

$$J_Q(\omega_x) = \frac{1}{2}\mathbb{E}_{s_t,a_t\sim\mathcal{D}}\Big[Q_{\omega_x}(s_t, a_t) - \hat{Q}(s_t, a_t)\Big]^2 \tag{6}$$

where the target $\hat{Q}(s_t, a_t) = r_t + \gamma\,\mathbb{E}_{\substack{s_{t+1}\sim p(\cdot|s_t,a_t)\\ a_{t+1}\sim\pi_\theta}}[\min_{x\in1,2} Q_{\bar{\omega}_x}(s_{t+1}, a_{t+1}) - \alpha\log\pi_\theta(a_{t+1}|s_{t+1})]$.

Furthermore, the policy parameters of standard SAC can be learned according to (5) by replacing $Q^{\pi_{\text{old}}}$ with current Q function estimate as

$$J_\pi(\theta) = \mathop{\mathbb{E}}_{s_t\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{a_t\sim\pi_\theta}\left[\alpha\log\pi_\theta(a_t|s_t) - \min_{x\in1,2} Q_{\omega_x}(s_t, a_t)\right]\right]. \tag{7}$$

Besides, since the gradient estimation of (7) has to depend on the actions stochastically sampled from π_θ, which leads to high-gradient variance, the reparameterization trick [37], [38] is used to transform the action generation process into deterministic computation, allowing for efficient gradient-based training. Concretely, the random action a_t can be expressed as a reparameterized variable

$$a_t = f_\theta(s_t; \delta_t) = \tanh(\mu_\theta(s_t) + \delta_t \odot \sigma_\theta(s_t)) \tag{8}$$

where $\odot$ represents Hadamard product, δ_t is sampled from $\mathcal{N}(0, \mathbf{I}_{\dim\mathcal{A}})$, and the mean μ_θ and standard σ_θ are outputs from the policy network parameterized by θ. This reparameterization enables the action a_t to be a differentiable function of θ, facilitating gradient descent methods.

In addition to the soft Q-function and the policy, the temperature parameter α can be automatically updated by optimizing the following loss:

$$J(\alpha) = \mathop{\mathbb{E}}_{s_t\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{a_t\sim\pi_\theta}\left[-\alpha\log\pi_\theta(a_t|s_t) - \alpha\bar{\mathcal{H}}\right]\right] \tag{9}$$

where $\bar{\mathcal{H}}$ is an entropy target with default value $-\dim\mathcal{A}$. Thus, the policy can explore more in regions where the optimal action is uncertain, and remain more deterministic in states with a clear distinction between good and bad actions. Besides, since two-timescale updates, i.e., less frequent policy updates, usually result in higher quality policy updates, we integrate the delayed policy update mechanism employed in TD3 [63] into our framework's methodology. To this end, the policy, temperature and target Q networks are updated with respect to soft Q network every e iterations.

2) Communication-Assisted Mixing Phase: Subsequent to the phase of local learning, neighboring agents initiate periodic communication to enhance their collaboration in accomplishing complex tasks. The messages transmitted by agents are limited to policy parameters θ as in many works [19], [20], [24]. Such an assumption is feasible as soft Q-functions have less impact on action selection than the policy in actor–critic algorithms.

Specifically, every U times of policy updates, a communication round begins. Agent i receives the policy parameters $\theta^{(j)}$ from neighboring agents $j \in \Omega_i$ via the D2D collaboration channel. Subsequently, agent i could employ various methods to formulate a referential policy $\tilde{\pi}^{(i)}$, which is parameterized by $\tilde{\theta}^{(i)} = f(\theta^{(1)}, \ldots, \theta^{(j)}, \ldots)$, that is, the parameters obtained from its neighbors $\forall j \in \Omega_i$. Afterward, agent i directly mixes DNN parameters of the policy network, consistently with parallel distributed SGD methods as

$$\theta_{\text{mix}}^{(i)} = \theta^{(i)} + \zeta\left(\tilde{\theta}^{(i)} - \theta^{(i)}\right) \tag{10}$$

where $\zeta \in [0, 1]$ is the mixture metric of DNN parameters. Taking model averaging in [30] and [39] as the example, $\tilde{\theta}^{(i)}$ is computed as $\tilde{\theta}^{(i)} = (1/|\Omega_i|) \sum_{j\in\Omega_i} \theta^{(j)}$, and $\zeta = 1 - 1/(|\Omega_i| + 1)$ is further influenced by the number of neighbors involved. Then, for each agent i, $\theta^{(i)}$ should get aligned with mixed policy's parameters $\theta_{\text{mix}}^{(i)}$.

B. Problem Formulation

This article primarily targets the communication-assisted mixing phase. Instead of simply continuing to follow the idea of the direct average of the DNN parameters in FL, an effective parameter mixture method could better leverage the exchanged parameters to yield a superior target referential policy. This approach aims to improve RL policy performance during the communication-assisted mixing phase, resulting in a consistently higher value with respect to the maximum entropy objective, as described in (1). However, it remains little investigated on the feasible means to mix the exchanged parameters (or their partial segments) and determine the proper mixture metric in (10), though it vitally affects both the communication overhead and learning performance. Therefore, by optimizing the mixture metric ζ, we mainly focus on reducing the communication expenditure while maintaining acceptable cumulative rewards, that is

$$\begin{aligned}
&\min_{\zeta} c(\upsilon, f) \\
&\text{s.t.} \sum_{t} r_t(\Theta, \zeta) \geq r^{\text{thre}} \\
&\Theta \leftarrow \left\{\theta_{\text{mix},k}^{(1)}, \ldots, \theta_{\text{mix},k}^{(N)}\right\} \\
&\theta_{\text{mix},k}^{(i)} = \theta_k^{(i)} + \zeta\left(\tilde{\theta}_k^{(i)} - \theta_k^{(i)}\right) \ \forall i \in \{1, \ldots, N\} \\
&\tilde{\theta}_k^{(i)} = f\left(\theta_k^{(1)}, \ldots, \theta_k^{(j)}, \ldots\right) \ \forall k \bmod U = 0, j \in \Omega_i
\end{aligned} \tag{11}$$

where r^{thre} denotes the required minimum cumulative rewards, v indicates the transmission bits of policy parameters, and k is the index of policy iterations. Furthermore, $c(v, f)$ denotes the communication expenditure, governed by the utilization function f, which represents the method of utilizing the parameter or parameter segments obtained from different neighboring agent $j \in \Omega_i$ to construct a referential policy parameterized by $\tilde{\theta}$. The practical implementation of f can be contingent on various factors, i.e., the underlying communications capability of agents to simultaneously receive and handle signals and the environmental conditions. After obtaining the referential policy parameter $\tilde{\theta}$, it is worthwhile to resort to a more comprehensive design of ζ to calibrate the communicating agents and contents as well as regulate the means to mix parameters or parameter segments, so as to provide a guarantee of performance improvement.

IV. Mixed Performance Improvement Bound Theorem of FRL Communication Under MERL

Given the potential for significant variability in policy performance due to differences in training samples among multiple agents, as well as the staleness of iterations caused by the varying computing power of different agents, it is critical to recognize that not all referential policies—those amalgamated with the agent's own policy via the DNN parameter mixture approach detailed in (10)—can contribute positively to agent's local learning process. More seriously, it may even degrade the learning performance sometimes [39], [41]. Therefore, in order to ensure the policy improvement post-mixture, a robust theoretical analysis method for evaluating the efficacy of the mixed policy distribution π_{mix}, approximated by DNN with parameter θ_{mix}, is essential. This would regulate the selection of only those referential policies $\tilde{\pi}$, parameterized by $\tilde{\theta}$, that are beneficial, in the parameter mixture process.

Drawing from this premise, based upon the conservative policy iteration as outlined in [40], we employ a mixture update rule on policy distributions to find an approximately optimal policy. Our analysis here will focus specifically on calibrating the mixing means of policy distributions π that exhibit a monotonic policy improvement property during the communication-assisted mixing phase. Notably, due to the nonlinear transformation in DNN and possible applicability of Softmax or restricted reparameterization [64], the mixture of policy distribution in (12) is not directly equivalent to the mixture of their parameters, especially when the parameter mixture takes the form in (10) under distributed SGD. Hence, the detailed and more practical mixture implementation for the policy network parameter θ, as in (10), will be derived from this section and more thoroughly presented in Section V-A2.

For any state s, we also define the mixed policy π_{mix}, which refers to a mixed distribution, as the linear combination of any referential policy $\tilde{\pi}$ and current policy π

$$\pi_{\text{mix}}(a|s) = (1 - \beta)\pi(a|s) + \beta\tilde{\pi}(a|s) \tag{12}$$

where $\beta \in [0, 1]$ is the weighting factor. The soft policy improvement, as outlined in Appendix-A, suggests that this policy mixture can influence the ultimate performance regarding the objective stated in (1). Fortunately, we have the following new theorem on the performance gap associated with adopting these two different policies.

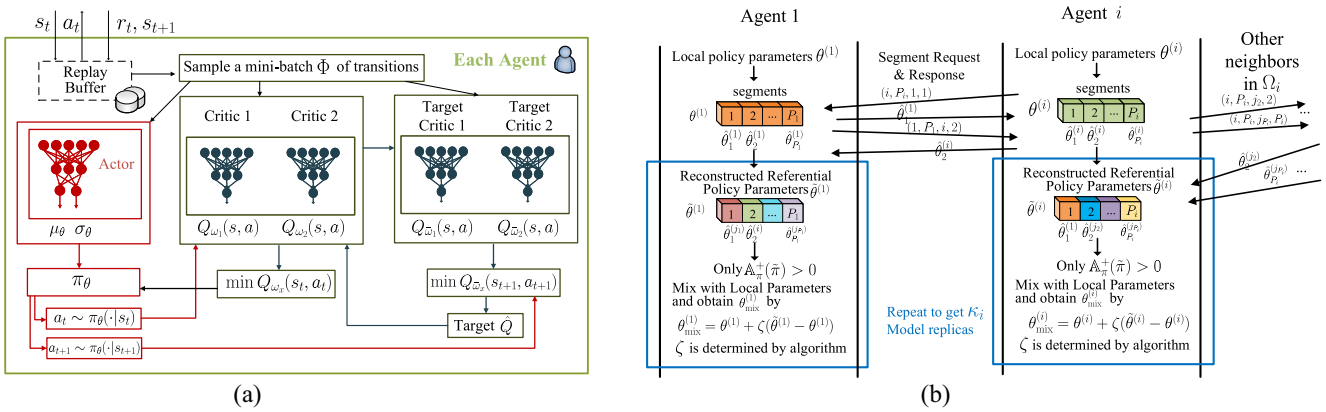


Fig. 2. Illustration of RSM-MASAC implementation. (a) Independent local learning phase. (b) Communication-assisted mixing phase.

Theorem 1 (Mixed Policy Improvement Bound): For any policy π and $\tilde{\pi}$ adhering to (12), the improvement in policy performance after mixing can be measured by

$$\eta(\pi_{\text{mix}}) - \eta(\pi) \geq \beta \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\tilde{\pi}}} \left[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s))\right] - \frac{2\gamma\varepsilon\beta^2}{(1-\gamma)^2} + \alpha \mathop{\mathbb{E}}_{s\sim d_{\pi_{\text{mix}}}} \left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right] \tag{13}$$

where $\varepsilon := \max_s |\mathbb{E}_{a\sim\tilde{\pi}}[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s))]|$ represents the maximum advantage of $\tilde{\pi}$ relative to π and $D_{\text{JS}}^{\beta}(p\|q) = \beta\sum p\log(p/[\beta p + (1-\beta)q]) + (1-\beta)\sum q\log(q/[\beta p + (1-\beta)q])$ is the β-skew Jensen–Shannon (JS)-symmetrization of KL divergence [65], with two distributions p and q.

The proof of this theorem is given in Appendix-B.

Remark: Notably, the proof effectively tackles the difficulties arising from the redefined, soft state value function with the extra logarithmic term of the policy by utilizing entropy decomposition of $H(\pi_{\text{mix}}(\cdot|s))$ in Lemma 5 as well as several mathematical tricks. Hence, this sets the stage for theoretically evaluating the performance of the mixed policy before the mixture occurs. The mixed policy improvement bound in (13) implies that under the condition that the right-hand side of (13) is larger than zero, the mixed policy will assuredly lead to an improvement in the true expected objective η. Besides, from another point of view, any mixed policy with a guaranteed policy improvement in Theorem 1 definitely satisfies the soft policy improvement as well, since the item $\mathbb{E}_{s_1}(\alpha H(\pi_{\text{old}}(\cdot|s_1)) + \mathbb{E}_{a_1\sim\pi_{\text{old}}}[Q^{\pi_{\text{old}}}(s_1,a_1)]) = \mathbb{E}_{s_1}[V^{\pi_{\text{old}}}(s_1)]$ in (22) in the Appendix-A (i.e., proof of Lemma 1) is less than $\mathbb{E}_{s_1}[V^{\pi_{\text{mix}}}(s_1)]$. In addition, when the temperature parameter $\alpha = 0$, this inequality can reduce to the standard form of policy improvement in traditional RL [39], [40], [42]. Hence, Theorem 1 derives a more general conclusion for both MERL and traditional RL.

Furthermore, since for $\beta \in [0,1]$, D_{JS}^{β} is greater than zero [65], only the sign of the first two terms in (13) needs to be considered. By applying the redefined policy advantage under MERL

$$\mathbb{A}_\pi^+(\tilde{\pi}) := \mathop{\mathbb{E}}_{s\sim d_\pi, a\sim\tilde{\pi}} \left[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s))\right] \tag{14}$$

we can therefore establish a tighter yet more tractable bound for policy improvement as

$$\eta(\pi_{\text{mix}}) - \eta(\pi) \geq \beta\mathbb{A}_\pi^+(\tilde{\pi}) - C\beta^2 \tag{15}$$

where $C = ([2\varepsilon\gamma]/[(1-\gamma)^2])$. Thus, (15) indicates that a mixed policy conforms to the principle of soft policy improvement, provided that the right side of (15) yields a positive value. More specifically, if the policy advantage $\mathbb{A}_\pi^+(\tilde{\pi})$ is positive, an agent with policy π can reap benefits by mixing its policy distribution with referential policy $\tilde{\pi}$. On the contrary, if this advantage is nonpositive, policy improvement cannot be assured through the mixing of $\tilde{\pi}$ and π. In essence, (15) serves as a criterion for selecting referential policies that ensure final performance improvement, and thus we have the following corollary.

Corollary 1: To obtain guaranteed performance improvement, the mixture approach of policy distributions shall satisfy that 1) $\mathbb{A}_\pi^+(\tilde{\pi}) > 0$ and 2) $\beta\mathbb{A}_\pi^+(\tilde{\pi}) - C\beta^2 > 0$.

V. MASAC With Regulated Segment Mixture

In this section, as shown in Fig. 2, we present the design of RSM-MASAC, which reduces the communication overhead while incurring little sacrifice to the learning performance.

A. Algorithm Design

Consistent with the SAC setting as in Section III, agents in RSM-MASAC undergo the same local iteration process. Meanwhile, for the communication-assisted mixing phase, we will elaborate on the design details of RSM-MASAC, including the segment request and response and policy parameter mixture with theory-established performance improvement.

1) Segment Request and Response: Inspired by segmented pulling synchronization mechanism in DFL [66], we develop and perform a segment request and response procedure. The proposed approach divides transmission of the policy parameters into segments, and each agent selectively requests various segments of policy parameters from different neighbors simultaneously through D2D communication, thereby facilitating the construction of a reconstructed referential policy for subsequent aggregation while also effectively balancing the load of communication costs and optimizing bandwidth usage.

Specifically, for every communication round, each agent i breaks its policy parameters $\theta^{(i)}$ into P_i ($P_i = \min\{P, |\Omega_i|\}$, according to the default segmentation granularity P and the number of neighbors $|\Omega_i|$ in current time) nonoverlapping segments $\hat{\theta}_1^{(i)}, \hat{\theta}_2^{(i)}, \ldots, \hat{\theta}_{P_i}^{(i)}$ as

$$\theta^{(i)} = \left(\hat{\theta}_1^{(i)}, \hat{\theta}_2^{(i)}, \ldots, \hat{\theta}_{P_i}^{(i)}\right). \tag{16}$$

Significantly, the available segmentation strategies are diverse and include, but are not limited to, dividing the policy parameters by splitting DNN layers [33], modular approach [67], or segmenting according to the parameter size [66]. Without loss of generality, taking IoV as an application example, there exist rather diverse V2V communication protocols and technologies, such as DSRC (based on IEEE 802.11p and its subsequent IEEE 802.11bd), NR-V2X (as specified in 3GPP Release 16 and 17), which support multiple input–multiple output (MIMO) technology and advanced signal processing techniques, such as spatial multiplexing and beamforming. Therefore, the communication capability depends on multiple factors ranging from physical layer configurations, such as antenna array setup and channel conditions, to upper layer protocols, shaping the theoretical upper limit for segmentation granularity $P_{\max}$. Besides, shorter distances between agents (e.g., vehicles) and better channel conditions generally enhance the capacity and reliability of the system. Still, the segmentation also can be dynamic, with larger $P_{\max}$ in environments with higher agent density and smaller $P_{\max}$ in cases where fewer agents are within communication range.

To clarify this process, we use the most intuitive uniform parameter partition. For each segment $p = 1, \ldots, P_i$, agent i randomly selects a target agent (without replacement) from its neighbors (i.e., $j_p \in \Omega_i$) to send segment request (i, P_i, j_p, p), which indicates the agent i who initiates the request, its total segment number P_i, as well as the requested segment p from the target agent j_p. Upon receiving the request, the agent j_p will break its own policy parameters $\theta^{(j_p)}$ into P_i segments and return the corresponding requested segment $\hat{\theta}_p^{(j_p)}$ according to the identifier p. Then, agent i could reconstruct a referential policy based on all of the fetched segments, that is

$$\tilde{\theta}^{(i)} = \left(\hat{\theta}_1^{(j_1)}, \hat{\theta}_2^{(j_2)}, \ldots, \hat{\theta}_{P_i}^{(j_{P_i})}\right). \tag{17}$$

In fact, this segmented transmission approach can be executed in parallel, thereby optimizing the utilization of available bandwidth. Instead of being confined to a single link, the traffic is distributed across P_i links, enhancing the overall data transfer efficiency. Besides, in order to further accelerate the propagation and ensure the model quality, we can construct multiple model replicas in RSM-MASAC. That is, the process of segment request and response can be repeated $P_i \times \kappa_i$ times, reconstructing $\kappa_i = \min\{\kappa, |\Omega_i|\}$ reconstructed referential policies in one communication round.

2) Policy Parameter Mixture With Theory-Established Performance Improvement: Consistent with TRPO [42], we introduce KL divergence to replace β by setting $\beta := \sqrt{D_{\mathrm{KL}}^{\max}(\pi \| \pi_{\mathrm{mix}})}$, where $D_{\mathrm{KL}}^{\max}(\pi \| \pi_{\mathrm{mix}}) = \max_s D_{\mathrm{KL}}(\pi(\cdot|s) \| \pi_{\mathrm{mix}}(\cdot|s))$. Thus, the second condition in Corollary 1 is equivalent to

$$\sqrt{D_{\mathrm{KL}}^{\max}(\pi \| \pi_{\mathrm{mix}})} < \frac{\mathbb{A}_\pi^+(\tilde{\pi})}{C}. \tag{18}$$

Nevertheless, it hinges on the computation-costly KL divergence to quantify the difference between probability distributions. Fortunately, since for a small change in the policy parameters, the KL divergence between the original policy and the updated policy can be approximated using a second-order Taylor expansion, wherein FIM serves as the coefficient matrix for the quadratic term. This provides a tractable way to assess the impact of parameter changes on the policy. Therefore, we utilize FIM, delineated in context of natural policy gradients by [68], as a mapping mechanism to revise the impact of certain changes in policy parameter space on probability distribution space. Then in the following theorem, we can get the easier-to-follow, trustable upper bound for the mixture metric of policy DNN parameters.

Theorem 2 (Guaranteed Policy Improvement via Parameter Mixing): With any referential policy parameters $\tilde{\theta}$, an agent with current policy parameters θ can improve the true objective η as in (1) through updating θ to mixed policy parameters θ_{mix} in accordance with (10), provided it fulfills the following two conditions.

1) $\mathbb{A}_\pi^+(\tilde{\pi}) > 0$.
2) $0 < \zeta < [(2\mathbb{A}_\pi^+(\tilde{\pi}))/(C[(\tilde{\theta} - \theta)^\intercal F(\theta)(\tilde{\theta} - \theta)])]^{1/2}$.

where $F(\theta)$ is the FIM of policy π parameterized by θ.

Proof: Recalling the definition of β and the mixture approach of θ_{mix} in (10), as well the change of policy parameters (i.e., $\Delta\theta = \zeta(\tilde{\theta} - \theta)$), for any state s, the KL divergence between the current policy and the mixed policy can be expressed by performing a second-order Taylor expansion of the KL divergence at the point θ in parameter space as

$$\begin{aligned}
&D_{\mathrm{KL}}(\pi_\theta(\cdot|s) \| \pi_{\theta+\Delta\theta}(\cdot|s)) \\
&= \sum_{a \in \mathcal{A}} \pi_\theta(a|s) \log \frac{\pi_\theta(a|s)}{\pi_{\theta+\Delta\theta}(a|s)} \\
&\approx D_{\mathrm{KL}}(\pi_\theta(\cdot|s) \| \pi_\theta(\cdot|s)) - \mathop{\mathbb{E}}_{a \sim \pi_\theta}\left[\frac{\partial \log \pi_\theta(a|s)}{\partial \theta}\right]^\intercal \Delta\theta \\
&\quad + \frac{1}{2}\Delta\theta^\intercal F(\theta)\Delta\theta \\
&\overset{(a)}{=} \frac{1}{2}\zeta^2\left(\tilde{\theta} - \theta\right)^\intercal F(\theta)\left(\tilde{\theta} - \theta\right)
\end{aligned}$$

where the equality (a) comes from the fact that by definition, and $D_{\mathrm{KL}}(\pi_\theta(\cdot|s) \| \pi_\theta(\cdot|s)) = 0$, while $\mathbb{E}_{a \sim \pi_\theta}\left[\partial \log \pi_\theta(a|s)/\partial\theta\right] = \sum_{a \in \mathcal{A}}\left[\pi_\theta(a|s)\partial \log \pi_\theta(a|s)/\partial\theta\right] = \sum_{a \in \mathcal{A}}\left[\partial \pi_\theta(a|s)/\partial\theta\right] = 0$. Notably, the FIM $F(\theta)$ takes the expectation for all possible states, which reflects the average sensitivity of the whole state space rather than a particular state, and can be calculated as

$$F(\theta) = \mathop{\mathbb{E}}_{\substack{s \sim d_\pi \\ a \sim \pi_\theta}}\left[\left(\frac{\partial \log \pi_\theta(a|s)}{\partial \theta}\right)\left(\frac{\partial \log \pi_\theta(a|s)}{\partial \theta}\right)^\intercal\right].$$

Finally, combining Corollary 1 and (18), we have the theorem. ■

Remark: With Theorems 1 and 2, we can anticipate the extent of policy performance changes resulting from policy DNN parameters mixture during the communication-assisted mixing phase. This prediction is based solely on the local curvature of the policy space provided by FIM, thus avoiding the cumbersome computations to derive the full KL divergence of two distributions for every state. Besides, given that $F(\theta)$ is a positive definite matrix, a positive sign of the policy advantage $\mathbb{A}_{\pi}^{+}(\tilde{\pi})$ implies an increase in the DNN parameters' mixture metric ζ corresponding to the rise in $\mathbb{A}_{\pi}^{+}(\tilde{\pi})$. That is, rather than simple averaging, agents can achieve guaranteed soft policy improvement after mixing by learning more effectively from referential policies with an elevated ζ.

In practice, to evaluate $\mathbb{A}_{\pi}^{+}(\tilde{\pi})$ and $F(\theta)$, the expectation can be estimated by the Monte Carlo method, approximating the global average by the states and actions under the policy. Meanwhile, the importance sampling estimator is also adopted to use the off-policy data in the replay buffer for the policy advantage estimation, where π_t typically denotes the action sampling policy at time step t. As outlined in Lemma 7 in Appendix-C, we have

$$\mathbb{A}_{\pi}^{+}(\tilde{\pi}) \approx \mathbb{E}_{s_t,a_t\sim\mathcal{D}}\Bigg[\left(\frac{\tilde{\pi}_{\tilde{\theta}}(a_t|s_t)-\pi_{\theta}(a_t|s_t)}{\pi_t(a_t|s_t)}\right)\min_{x\in 1,2} Q_{\omega_x}(s_t,a_t) + \alpha\Big[H\big(\tilde{\pi}_{\tilde{\theta}}(\cdot|s_t)\big)-H(\pi_{\theta}(\cdot|s_t))\Big]\Bigg] \tag{19}$$

and

$$F(\theta) \approx \mathop{\mathbb{E}}_{s_t\sim\mathcal{D}}\Bigg[\mathop{\mathbb{E}}_{a_t\sim\pi_{\theta}}\Bigg[\left(\frac{\partial \log \pi_{\theta}(a|s)}{\partial\theta}\right)\left(\frac{\partial \log \pi_{\theta}(a|s)}{\partial\theta}\right)^{\mathsf{T}}\Bigg]\Bigg]. \tag{20}$$

Finally, we summarize the details of RSM-MASAC in Algorithm 1. RSM-MASAC employs a theory-guided metric for policy parameter mixture, which takes into account the potential influence of parameters mixture on soft policy improvement under MERL—a factor commonly overlooked in parallel distributed SGD methodologies. In particular, the mixing process between any two agents is initiated solely when there is a positive policy advantage. Adhering to Theorem 2, the mixture metric is set marginally below its computed upper limit, ensuring both policy improvement and convergence.

Algorithm 1 RSM-MASAC Algorithm

Initialize network parameters $\theta^{(i)}$, $\omega_1^{(i)}$, $\omega_2^{(i)}$, $i = 1, 2, \ldots N$.
Initialize target network parameters $\bar{\omega}_1^{(i)} \leftarrow \omega_1^{(i)}$, $\bar{\omega}_2^{(i)} \leftarrow \omega_2^{(i)}$.
Initialize learning rate η_π, η_Q, η_α, temperature α, communication interval U, segmentation granularity P, predefined replicas κ.
Initialize iteration index $k \leftarrow 0$ and *counter* $\leftarrow 0$.
for each epoch **do**
  for $t \leftarrow 1$ to T **do**
    Each agent i **executes:**
    *** Independent local learning phase**
    Select an action $a_t^{(i)}$ with respect to $s_t^{(i)}$ according to the current policy $\pi^{(i)}$ parameterized by $\theta^{(i)}$.
    Observe reward $r_t^{(i)}$ and next state $s_{t+1}^{(i)}$.
    Save the new transition in replay buffer: $\mathcal{D}^{(i)} \leftarrow \mathcal{D}^{(i)} \cup \langle s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)} \rangle$.
    Sample a mini-batch $\Phi^{(i)} \sim \mathcal{D}^{(i)}$.
    Update soft Q function $\omega_x^{(i)} \leftarrow \omega_x^{(i)} - \eta_Q \nabla J_Q(\omega_x^{(i)})$ by (6), for $\forall x = 1, 2$.
    if *counter* mod $d = 0$ **or** $t = T$ **then**
      Update policy $\theta_{k+1}^{(i)} \leftarrow \theta_k^{(i)} - \eta_\pi \nabla J_\pi(\theta_k^{(i)})$ by (7).
      Adjust temperature $\alpha^{(i)} \leftarrow \alpha^{(i)} - \eta_\alpha \nabla J(\alpha^{(i)})$ by (9).
      Update target networks $\bar{\omega}_x^{(i)} \leftarrow \varrho\omega_x^{(i)} + (1-\varrho)\bar{\omega}_x^{(i)}$, for $\forall x = 1, 2$.
      $k \leftarrow k + 1$.
    end if
    counter $\leftarrow$ *counter* $+ 1$.
    *** Communication-assisted mixing phase**
    if k mod $U = 0$ **then**
      Update policy $\theta_k^{(i)} \leftarrow$ **CommMix**$(\theta_k^{(i)}, i, \Omega_i, P, \kappa)$ according to Algorithm 2.
    end if
  end for
end for

B. Communication Cost Analysis

Since the pulling request does not contain any actual data, its cost in the analysis can be ignored, and we only consider the policy parameters transmitted among agents to analyze the communication efficiency of RSM-MASAC.

Regarding the communication overhead for each segment request, RSM-MASAC incurs a maximization data transmission cost of $\upsilon_p = \upsilon/P$ through D2D communications. Following the same settings as in [67], we provide illustrative IoV examples to demonstrate the overall reduction in communication overhead. Specifically, the collective perception message (CPM) defined in collective perception service (CPS) of ETSI TR 103 562 [69] can encapsulate DNN parameters into the perceived object containers (POCs) and propagate them among cooperating DRL agents with each message carrying a payload of 4480 bytes, serving as segment responses in our algorithm. And the DNN parameters $\theta \in \mathbb{R}^d$'s transmission size υ can be regarded as $32d$ bits, commonly assumed in [25], [58], and [67]. Consequently, the total communication overhead for reconstructing maximal κ referential policies per agent in each round $c(\upsilon, f) = ([N \times \upsilon \times \kappa]/[8 \times 1024^3])$ (GB), as well as $([N \times \upsilon \times \kappa]/[8 \times 4480])$ message numbers, which is $(N-1)/\kappa$ times less than that in a fully connected communication setup [39]. Additionally, communication occurs at a periodic interval of U, allowing for further reduction in communication overhead by decreasing communication frequency. Moreover, by simultaneously requesting P_i agents in parallel, RSM-MASAC benefits from the sufficient use of the bandwidth and enhances the capability to overcome possible channel degradation.

Algorithm 2 *CommMix* Function in Algorithm 1

Input: θ, i, Ω_i, P, κ.
$P_i = \min\{P, |\Omega_i|\}$ and $\kappa_i = \min\{\kappa, |\Omega_i|\}$.
for each replica $1, 2, \ldots, \kappa_i$ **do**
  Send P_i pulling request (i, P_i, j_p, p) to nearby collaborators in Ω_i, and receive $\hat{\theta}_p^{(j_p)}$ to reconstruct $\tilde{\theta}$ as (17).
  Select M samples from the replay buffer $\mathcal{D}^{(i)}$.
  Estimate $\mathbb{A}_\pi^+(\tilde{\pi})$ according to (19).
  if $\mathbb{A}_\pi^+(\tilde{\pi}) > 0$ **then**
    Evaluate $F(\theta)$ according to (20).
    Get the upper bound of ζ according to Theorem 2.
    Make the mixture metric ζ less than the calculated upper bound, and update $\theta_{\text{mix}}^{(i)}$ by (10).
  end if
end for
Output: $\theta_{\text{mix}}^{(i)}$.

C. Complexity Analysis

For each referential policy, the calculation of policy advantage $\mathbb{A}_\pi^+(\tilde{\pi})$ is contingent with sample size M, policy parameter size $|d|$ and Q network parameter size $|d_Q|$, with complexity $\mathcal{O}(M(|d| + |d_Q|))$. However, the mixture of policy parameters only proceeds if condition $\mathbb{A}_\pi^+(\tilde{\pi}) > 0$ is met during the communication-assisted mixing phase, with the computation complexity primarily depending on the calculation of FIM. Specifically, with sampling approximated method using first-order gradients in (20), the computation complexity of FIM is $\mathcal{O}(M|d|^2)$ and the associated memory complexity is $\mathcal{O}(|d|^2)$. However, the actual computation of the FIM is not restricted to sampling-based method alone, there has been extensive research on FIM approximation methods, including diagonal approximation [70], [71] and low-rank approximation [72], among others, to further reduce to the complexity.

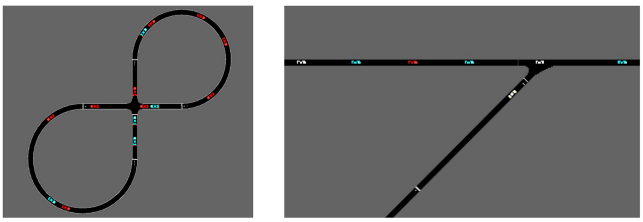
VI. Experimental Results and Discussions

In this section, we validate the effectiveness of our proposed algorithm for the speed control of connected automated vehicles (CAVs) in IoV, highlighting its superiority compared to other methods.

A. Experimental Settings

We implement two simulation scenarios on Flow [73], [74], which is a traffic control benchmarking framework for mixed autonomy traffic. As illustrated in Fig. 3, the common urban traffic intersection scenario "Figure 8" and highway scenario "Merge" are selected, with the main system settings described in Table III.

1) *Figure 8:* 14 vehicles navigate a one-way lane shapes like a figure "8", including five emulated human driven vehicles (HDVs), controlled by simulation of urban mobility (SUMO) with intelligent driver model (IDM) [75], and nine IRL-controlled CAVs maintaining dedicated links to update their parameters through the V2V channel. At the lane's intersection, each CAV adjusts its acceleration to traverse efficiently, aiming to boost the traffic flow's average speed.
2) *Merge:* A highway on-ramp merging scenario with vehicle flow at 2300 per hour, including a maximum of 2200 vehicles on the main road and 100 vehicles on the ramp. Within each epoch, 13 vehicles are randomly chosen to instantiate the DRL-based controllers as they sequentially enter, aiming to manage collision avoidance and congestion at merge points. The simulation settings closely align with those of the first scenario.



(a) (b)

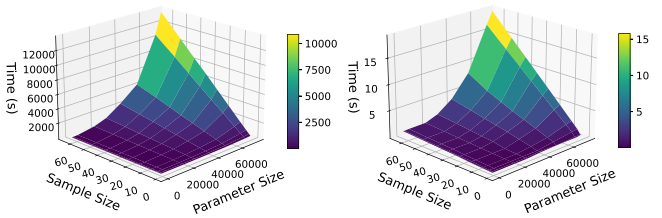
Fig. 3. Two scenarios for simulations on Flow. The red vehicles are DRL-driven CAVs, while the blue vehicles are the HDVs observed by the DRL-driven CAVs and the white vehicles are the HDVs that are not observed in the state space. (a) Figure 8. (b) Merge.

TABLE III
Setting of System Parameters in Two Scenarios

Parameters Definition	Figure 8	Merge
Number of DRL agents N	9	13
Total time-steps per epoch T	$1,500$	750
Number of epochs	300	260
Range of acceleration (m/s^2)	$[-3, 3]$	$[-1.5, 1.5]$
Desired velocity per vehicle (m/s)	20	20
Speed limit per vehicle (m/s)	30	30
Length per time-step (s)	0.1	0.1
Maximum of vehicles per hour	-	$2,300$

Both two scenarios are modified to assign the limited partial observation of the global environment as the state of each CAV, including the position and speed of its own, the vehicle ahead and behind. Only CAVs can execute the V2V end-to-end communication, and the connectivity of V2V links at time t depends on the CAVs' position and communication range, which are both extracted using the TraCI simulator in our experiments. A communication range of 90 m is adopted in interactions in Figure 8 and 400 m in highway merge. This is according to the conclusion in [76] that communication range at intersections will be extremely reduced compared with the conventional scenarios [77], and relevant service requirements [78]. Unless otherwise stated, our experiments are based on the common MARL V2V lossless ideal communication premise [7], [17], [27], [28]. Meanwhile, each CAV's action is a continuous variable representing speed acceleration or deceleration. It is sampled from the outputs of the policy network, which has three fully connected layers with 256 hidden units each and ReLU activations. Actions are squashed using a tanh function to fall within $[-1, 1]$ and then scaled by the acceleration bounds of the scenarios.

For each referential policy, parameter mixing occurs only when $\mathbb{A}_\pi^+(\tilde{\pi}) > 0$. In this case, the computational complexity



(a) (b)

Fig. 4. Runtime of estimating mixture metric under different sample and parameter sizes. (a) Runtime on CPU. (b) Runtime on GPU.

is predominantly governed by the calculation of $F(\theta)$ due to the outer product of policy gradients. Using an AMD EPYC 7F32 @3.70 GHz, 8-core CPU, and an NVIDIA RTX3090 GPU,[2] as shown in Fig. 4, the computational time escalates with increasing sample size and parameter size, consistent with the discussions in Section V-C. However, with the aid of GPU hardware acceleration, such as superior parallel processing capabilities, higher memory bandwidth, and optimized deep learning libraries, the runtime experiences a significant reduction.

In order to reduce the occurrence of collisions and promote the traffic flow to the maximum desired speed, we take the normalized average speed of all vehicles at each timestep as the individual reward in each scenario,[3] which is assigned to each training agent after its action is performed. In addition, the current epoch will be terminated once a collision occurs or the max length of step T in an epoch is reached.

The results with variance are averaged over 5 independent simulations, with tests conducted every ten epochs during training. Moreover, as for the baseline, we take the vehicles controlled by the flow IDM [75], which belongs to a typical car-following model incorporating extensive prior knowledge and indicates the pinnacle of performance achievable by the best centralized federated MARL algorithms [14]. Furthermore, the principal hyperparameters used in simulations are listed in Table IV.

B. Evaluation Metrics

Apart from the average reward, we adopt some additional metrics to extensively evaluate the communication efficiency of RSM-MASAC.

1) We denote the total count of reconstructed referential policies (i.e., all model replicas) as ρ_{total}. The number of effectively reconstructed referential policies, those contributing to the mixing process, is represented by ρ_{ef}. The mixing rate $\rho_r = \rho_{\text{ef}}/\rho_{\text{total}}$ thus reflects the usage rate of reconstructed policies.
2) We use ψ to indicate the overall communication overhead (in terms of υ) in an epoch, and C_0 denotes the number of communication rounds. Therefore, the communication overheads equal $\psi = C_0 \times c(\upsilon, f) = ([\rho_{\text{total}} \times \upsilon]/[8 \times 1024^3])$ (GB).

[2]We note that there exist available automotive chips like the NVIDIA DRIVE Orin or Thor featuring CUDA Tensor Core GPU.

[3]Notably, we assume complete knowledge of individual vehicle speeds at each vehicle here. Beyond the scope of this article, some value-decomposition methods like [79] can be further leveraged to derive a decomposed reward, so as to loosen such a strict requirement.

TABLE IV
HYPERPARAMETERS

Hyper-parameters	Symbol	Value
Replay buffer size	$\lvert\mathcal{D}\rvert$	10^5
Batch size	Φ	256
Number of samples to evaluate $\mathbb{A}_\pi^+(\tilde{\pi})$	M	50
Learning rate of actor network	η_π	4×10^{-5}
Learning rate of critic network	η_Q	3×10^{-4}
Learning rate of temperature parameter	η_α	3×10^{-4}
Discount factor	γ	0.99
Target smoothing coefficient	ϱ	10^{-3}
Delayed policy update intervals	e	10
Communication intervals	U	8
Segmentation granularity	P	4
Predefined replicas	κ	3

C. Simulation Results

Beforehand, we present the performance of independent SAC (ISAC) and IPPO for MARL without any information sharing in Fig. 5, so as to highlight the critical role of interagent communication in decentralized cooperative MARL and facilitate subsequent discussions. Fig. 5 reveals that the learning process of these noncooperative algorithms achieves unstable average reward with greater variance, and suffers from convergence issues, whereas when a communication-assisted mixing phase involving the exchange of policy parameters is incorporated, as shown in Fig. 6, the learning process within the multiagent environment is remarkably enhanced in terms of stability and efficiency. In other words, consistent with our previous argument, integrating DFL into IRL significantly improves training efficiency and ensures learning stability.

In Fig. 6, we also compare the performance of RSM and that of the direct, unselective Averaging (Avg) method, as mentioned in Section III. Note that to speed up the calculation, we use a batch average gradient to approximate the calculation of $F(\theta)$ here. In Fig. 6, it can be observed that in terms of convergence speed and stability, the simple average mixture method is somewhat inferior. Particularly in more complex scenarios, as in merge, which involves a greater number of RL agents and denser traffic flow compared to Figure 8, the disparity between these two methods becomes more pronounced. The larger variance in the average reward under the average mixture method suggests a more unstable parameter mixing process, while the improvement in reward value implies the efficiency of communication information sharing. Therefore, it validates the effectiveness of RSM and supports the derived theoretical results for selecting useful reference policies and assigning appropriate mixing weights.

On the other hand, Fig. 6 offers a further evaluation of the RSM's performance on both MASAC and MAPPO under the same policy DNN structure, learning rate and gradient clipping condition. In particular, as proposed in our previous work [1], RSM-MAPPO utilizes PPO, which features lower

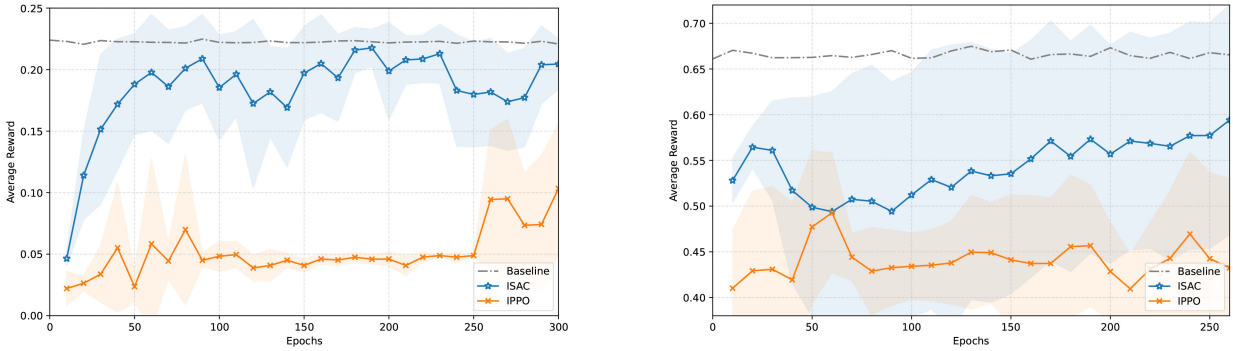


Fig. 5. Performance of IRL without communication. (a) Figure 8. (b) Merge.

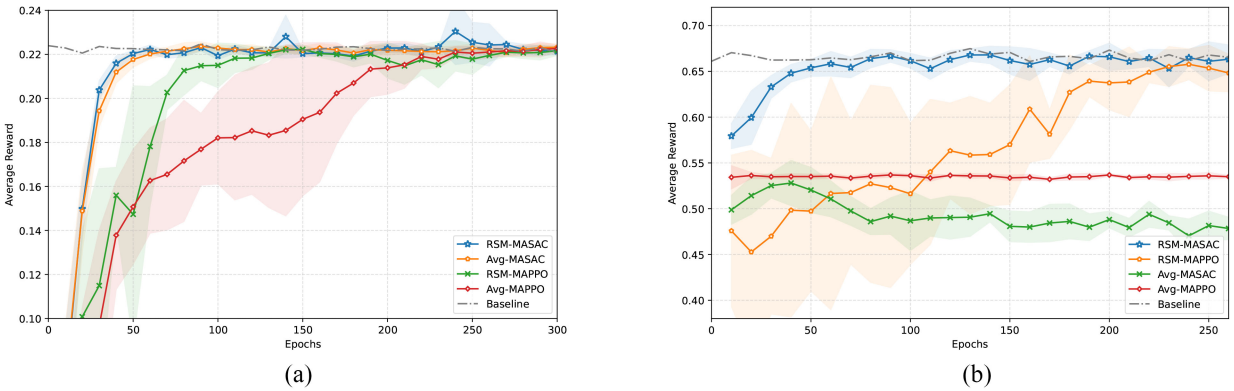


Fig. 6. Performance comparison of different methods. (a) Figure 8. (b) Merge.

sampling efficiency and policy update frequency, as the independent local learning algorithm under the traditional MARL framework, making it a special case of RSM-MASAC with $\alpha = 0$ and a different policy advantage estimation under our more general reanalyses. Evidently, in both scenarios, RSM-MASAC demonstrates faster convergence and overall superior performance compared to RSM-MAPPO. We believe this is primarily attributed to the differences in exploration mechanism and sample efficiency. Specifically, benefiting from the introduced entropy item under MERL framework, SAC encourages policies to better explore environments, thus capably avoiding local optima and potentially learning faster.

Next, we evaluate the impact of regulated segmentation. Fig. 7 shows that the incorporation of segmentation (i.e., $P = 4$) also leads to an improvement in the successful mixing rate ρ_r of the policy, apart from better utilizing the available bandwidth as discussed in Section V-B. The improvement lies in the introduction of a certain level of randomness, consistent with the idea to encourage exploration by adding an entropy term in MERL. But this randomness is also regulated by the policy parameter mixing theorem in Theorem 2 without compromising performance. To embody the selection process of reconstructed referential policies, we further present the heatmap comparison between communication times and successful mixture times in Fig. 8. It can be observed from Fig. 8(a) that agents communicate

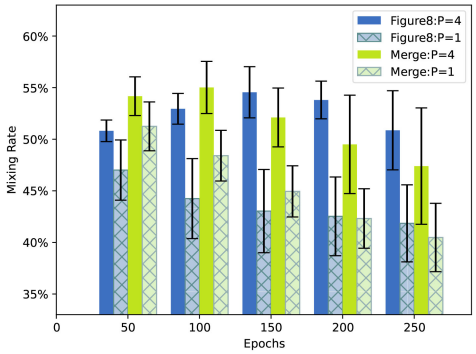


Fig. 7. Effect of segmentation on mixing rate.

frequently with their neighboring agents, but since not all communication packets (i.e., model segments) are utilized for improving local policy performance, it leads to the uneven and asymmetrical heatmap for the successful mixture of segments in Fig. 8(b). Our derived performance improvement bound and practical parameter mixture metric regulate these segments, demonstrating that the data from different agents contribute variably to the mixture.

In Fig. 9, we further examine the impact of varying the predefined segmentation granularity P and replicas κ under RSM-MASAC. Notably, for each agent, the actual segment number and replicas at each time are also affected by the

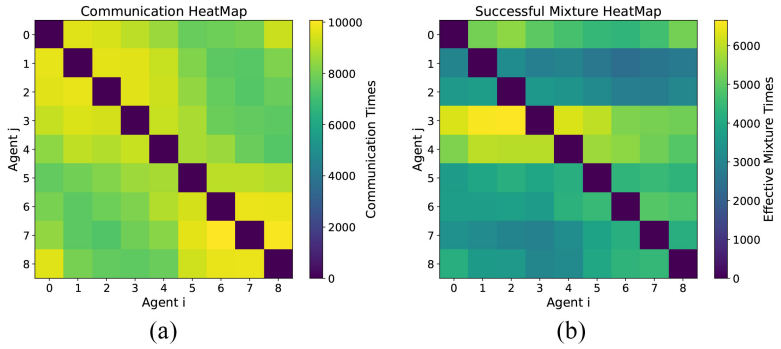


Fig. 8. Heatmap comparison of communication times and effective mixture times. (a) Communication times. (b) Effective mixture times.

TABLE V
RESULTS OF AVERAGE REWARD AND COMMUNICATION EFFICIENCY UNDER MERGE

Method		U	P	κ	Average Reward	ρ_{total}	ψ (GB)	ρ_{ef}	ρ_r
Avg-	MAPPO	8	4	3	0.5351 ± 0.0031	1,626	0.412	1,626	100%
	MASAC				0.4818 ± 0.0161	29,270	7.425	29,270	100%
RSM-	MAPPO				0.6528 ± 0.0199	1,546	0.392	711	45.99%
	MASAC			1	0.6538 ± 0.0250	10,223	2.593	5,213	50.99%
				5	0.6636 ± 0.0115	41,333	10.486	20,410	49.38%
				7	0.6574 ± 0.0156	38,120	9.670	19,722	51.74%
			3	3	0.6614 ± 0.0165	30,486	7.734	13,827	45.36%
			1		0.6535 ± 0.0253	30,066	7.627	12,117	40.30%
			2		0.6206 ± 0.0793	29,634	7.518	14,549	49.10%
			6		0.6505 ± 0.0282	30,126	7.643	13,902	46.15%
		72	4		0.6712 ± 0.0282	2,351	0.596	1,357	57.72%
		144			0.6576 ± 0.0269	1,227	0.311	708	57.70%

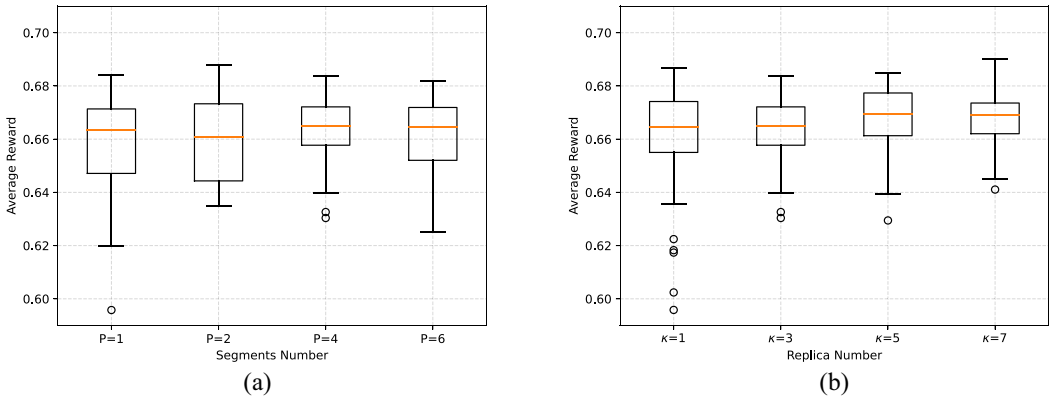


Fig. 9. Performance of RSM-MASAC with respect to different hyperparameters [(a) predefined number of segments P and (b) predefined number of replicas κ], based on the testing episodes after 130, 140, . . ., 200 training epochs under Merge.

number of neighbors in the communication range, as addressed in Section V-A. The boxplots display the distribution of average reward collected during testing epochs ranging from 130 to 200, a period marked by increased fluctuations and slower convergence. These outcomes are inherently subject to the stochastic nature of the environment interacting with the RL agents and here we only indicate a rough trend that the median average reward shows a modest increase with a greater number of segments in Fig. 9(a), and a slight upward trend with an increase replicas in Fig. 9(b). However, these variations are minimal and contingent upon specific environmental conditions that are not consistent, thus warrant further discussions to balance communication bandwidth and computational overhead in practical applications. In addition, we take the average reward within the last five testing epochs as the final converged performance, and summarize corresponding results and details about communication overhead in Table V, which takes the V2V communication examples' settings in Section V-B.

In addition, we conduct more ablation studies to evaluate the design of Q network on the learning process. In Fig. 10, we analyze the impact of different target smoothing coefficients ϱ on the double Q-learning process. It can be observed that a larger value of the target smoothing coefficient ϱ, such

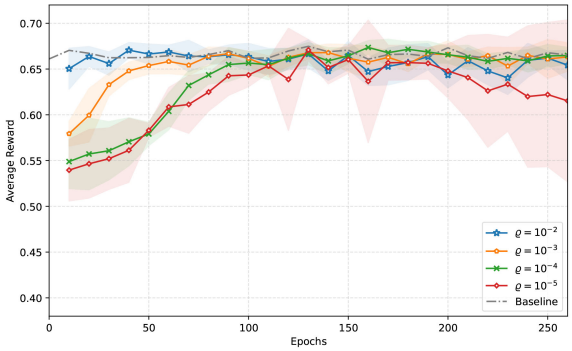


Fig. 10. Impact of different target smoothing coefficient ϱ under Merge.

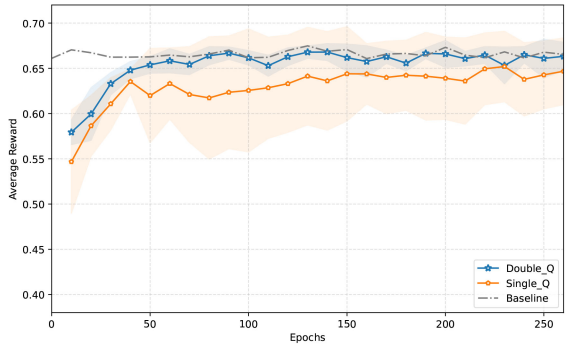


Fig. 11. Double versus single Q network under Merge.

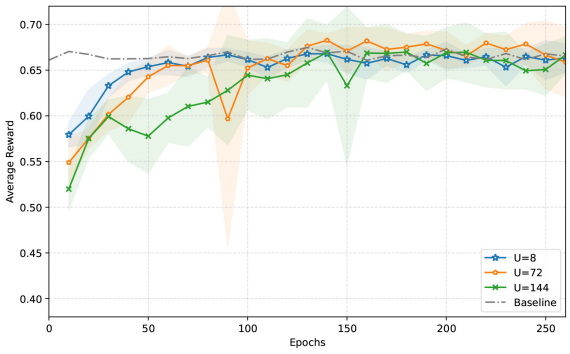


Fig. 12. Impact of different communication intervals U under Merge.

as 10^{-2}, can significantly accelerate the learning, but it also introduces disturbances, leading to nonsteady learning. On the other hand, a smaller value like 10^{-4} or 10^{-5} noticeably slows down the learning speed and also introduces instability, affecting the performance of the algorithm. Moreover, in Fig. 11, we also evaluate the adoption of dual Q networks. It can be observed that employing dual Q networks indeed accelerates the learning process. Furthermore, we also discuss the impact of communication intervals on the learning process, as shown in Fig. 12. It is evident that larger communication intervals result in reduced learning speeds and significant instability in the learning process, the same as classical FL processes.

Meanwhile, we also investigate the effect of model fine-tuning in decentralized IoV speed control settings. Specifically, in a new environment full of 14 DRL-controlled CAVs, we

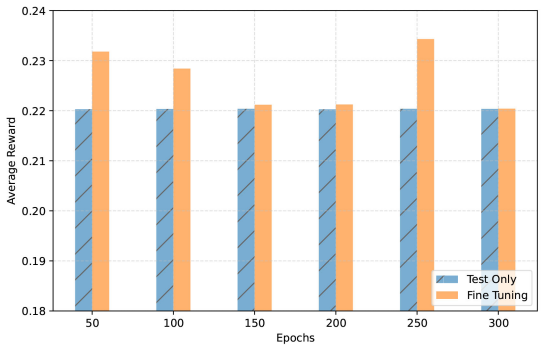


Fig. 13. Performance improvement of continually fine-tuning under Figure 8.

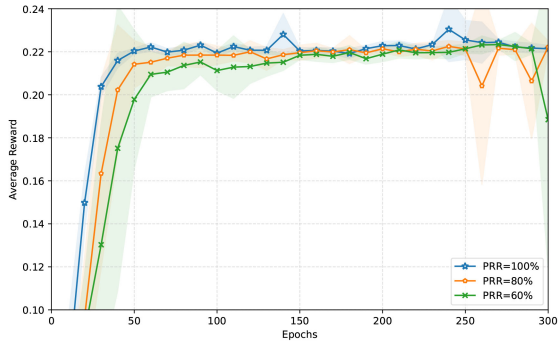


Fig. 14. Performance comparison of different PRR in communication under Figure 8.

compare the performance between a continually fine-tuning policy and a well-trained, convergent policy (i.e., RSM-MASAC in Fig. 6) to execute inference only (i.e., test only in DRL description). The results in Fig. 13 show that the fine-tuned policy in new conditions can reduce the occurrence of collisions at the beginning and continuously achieve better performance than just testing. Furthermore, to investigate the impact of possible packet loss when communicating on D2D or V2V links, we provide the result for different packet reception rate (PRR) in Fig. 14. It can be seen that a packet loss of less than 40% primarily affects the convergence speed at the beginning, but produces trivial differences on the converged performance, demonstrating that DNN parameter transmission can yield good robustness in most real communication environments.

VII. Conclusion and Future Work

In this article, we have proposed a communication-efficient algorithm RSM-MASAC as a promising solution to enhance communication efficiency and policy collaboration in distributed MARL, particularly in the context of highly dynamic environments. By delving into the policy parameter mixture function, RSM-MASAC has provided a novel means to leverage and boost the effectiveness of distributed multiagent collaboration. In particular, RSM-MASAC has successfully transformed the classical means of complete parameter exchange into segment-based request and response, which significantly facilitates the construction of multiple referential policies and simultaneously captures enhanced

learning diversity. Moreover, in order to avoid performance-harmful parameter mixture, RSM-MASAC has leveraged a theory-established regulated mixture metric, and selects the contributive referential policies with positive relative policy advantage only. Finally, extensive simulations in the mixed-autonomy traffic control scenarios have demonstrated the effectiveness of the proposed approach. Notably, we use an approximate calculation here to reduce the computation complexity and speed up the calculation of $F(\theta)$, thus supporting the validation of the mixed performance improvement bound theorem. Despite its efficiency, its impact is still under investigation, and other methods to simplify FIM computations can be explored and substituted in the future.

APPENDIX
PROOFS

A. Proof of Soft Policy Improvement

Lemma 1 (Soft Policy Improvement): Let π_{old} and π_{new} be the optimizer of the minimization problem defined in (5). Then $Q^{\pi_{\text{new}}}(s,a) \geq Q^{\pi_{\text{old}}}(s,a),\ \forall s,a$ with $|\mathcal{A}| < \infty$.

Proof: This proof is a direct application of soft policy improvement [37], [38]. We leave the proof here for completeness.

Let $\pi_{\text{old}} \in \Pi$ and Q^{π}, V^{π} is the corresponding soft state-action value and soft state value, respectively. And π_{new} is defined as

$$\begin{aligned}\pi_{\text{new}} &= \arg\min_{\pi\in\prod} D_{\text{KL}}\left(\pi(\cdot|s)\|\frac{\exp\left(\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s,\cdot)\right)}{Z^{\pi_{\text{old}}}(s)}\right)\\ &= \arg\min_{\pi\in\prod} J_{\pi_{\text{old}}}(\pi(\cdot|s)).\end{aligned}$$

Since we can always choose $\pi_{\text{new}} = \pi_{\text{old}} \in \Pi$, there must be $J_{\pi_{\text{old}}}(\pi_{\text{new}}(\cdot|s)) \leq J_{\pi_{\text{old}}}(\pi_{\text{old}}(\cdot|s))$. Hence

$$\begin{aligned}&\mathbb{E}_{a\sim\pi_{\text{new}}}\left[\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s,a) - \log\pi_{\text{new}}(a|s) \ - \log Z^{\pi_{\text{old}}}(s)\right]\\ &\geq \mathbb{E}_{a\sim\pi_{\text{old}}}\left[\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s,a) - \log\pi_{\text{old}}(a|s) \ - \log Z^{\pi_{\text{old}}}(s)\right].\end{aligned}$$

As partition function Z depends only on the state, the inequality reduces to a form of the sum of entropy and value with one-step look-ahead

$$\begin{aligned}&\mathbb{E}_{a\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(s,a)\right] + \alpha H(\pi_{\text{new}}(\cdot|s))\\ &\geq \mathbb{E}_{a\sim\pi_{\text{old}}}\left[Q^{\pi_{\text{old}}}(s,a)\right] + \alpha H(\pi_{\text{old}}(\cdot|s)) = V_{\pi_{\text{old}}}(s).\end{aligned}$$

And according to the definition of the soft Q-value in Section II, we can get that

$$\begin{aligned}&Q^{\pi_{\text{old}}}(s,a)\\ &= \mathop{\mathbb{E}}_{s_1}\left[r_0 + \gamma\left(\alpha H(\pi_{\text{old}}(\cdot|s_1)) + \mathop{\mathbb{E}}_{a_1\sim\pi_{\text{old}}}\left[Q^{\pi_{\text{old}}}(s_1,a_1)\right]\right)\right]\\ &\leq \mathop{\mathbb{E}}_{s_1}\left[r_0 + \gamma\left(\alpha H(\pi_{\text{new}}(\cdot|s_1)) + \mathop{\mathbb{E}}_{a_1\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(s_1,a_1)\right]\right)\right]\\ &= \mathop{\mathbb{E}}_{\substack{s_1\\ a_1\sim\pi_{\text{new}}}}\left[r_0 + \gamma\left(\alpha H(\pi_{\text{new}}(\cdot|s_1)) + r_1\right)\right]\\ &\quad + \gamma^2\mathop{\mathbb{E}}_{s_2}\left[\alpha H(\pi_{\text{old}}(\cdot|s_2)) + \mathop{\mathbb{E}}_{a_2\sim\pi_{\text{old}}}\left[Q^{\pi_{\text{old}}}(s_2,a_2)\right]\right]\\ &\leq \mathop{\mathbb{E}}_{\substack{s_1\\ a_1\sim\pi_{\text{new}}}}\left[r_0 + \gamma\left(\alpha H(\pi_{\text{new}}(\cdot|s_1)) + r_1\right)\right]\\ &\quad + \gamma^2\mathop{\mathbb{E}}_{s_2}\left[\alpha H(\pi_{\text{new}}(\cdot|s_2)) + \mathop{\mathbb{E}}_{a_2\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(s_2,a_2)\right]\right]\\ &= \mathop{\mathbb{E}}_{\substack{s_1\\ a_1\sim\pi_{\text{new}}\\ s_2}}\left[r_0 + \gamma\left(\alpha H(\pi_{\text{new}}(\cdot|s_1)) + r_1\right)\right.\\ &\qquad \left. + \gamma^2\left(\alpha H(\pi_{\text{new}}(\cdot|s_2)) + r_2\right)\right]\\ &\quad + \gamma^3\mathop{\mathbb{E}}_{s_3}\left[\alpha H(\pi_{\text{new}}(\cdot|s_3)) + \mathop{\mathbb{E}}_{a_3\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(s_3,a_3)\right]\right]\\ &\quad \cdots\\ &\leq \mathop{\mathbb{E}}_{\tau_1\sim\pi_{\text{new}}}\left[r_0 + \sum_{t=1}^{\infty}\gamma^t\left(H(\pi_{\text{new}}(\cdot|s_t)) + r_t\right)\right]\\ &= Q^{\pi_{\text{new}}}(s,a).\end{aligned} \tag{21}$$

■

B. Proof of Theorem 1

Before proving the theorem, we first introduce several important lemmas, on which the proofs of the proposed theorems are built.

Lemma 2:

$$\mathbb{E}_{a\sim\pi}[A_\pi(s,a)] = -\alpha H(\pi(\cdot|s)).$$

Proof:

$$\begin{aligned}&\mathbb{E}_{a\sim\pi}[A_\pi(s,a)]\\ &= \sum_a \pi(a|s)A_\pi(s,a)\\ &= \sum_a \pi(a|s)[Q_\pi(s,a) - V_\pi(s)]\\ &= \sum_a \pi(a|s)Q_\pi(s,a) - V_\pi(s)\\ &\overset{(a)}{=} \sum_a \pi(a|s)Q_\pi(s,a) - \sum_a \pi(a|s)\left[Q_\pi(s,a) - \alpha\log\pi(a|s)\right]\\ &= -\alpha H(\pi(\cdot|s))\end{aligned}$$

where the equality (a) is according to (4). ■

Lemma 3:

$$\mathop{\mathbb{E}}_{a\sim\pi_{\text{mix}}}[A_\pi(s,a)] = \beta\mathop{\mathbb{E}}_{a\sim\tilde{\pi}}[A_\pi(s,a)] - (1-\beta)\alpha H(\pi(\cdot|s)).$$

Proof:

$$\begin{aligned}&\mathbb{E}_{a\sim\pi_{\text{mix}}}[A_\pi(s,a)]\\ &= \sum_a \pi_{\text{mix}}(a|s)A_\pi(s,a)\\ &\overset{(a)}{=} \sum_a\left[(1-\beta)\pi(a|s) + \beta\tilde{\pi}(a|s)\right]A_\pi(s,a)\\ &= \beta\sum_a \tilde{\pi}(a|s)A_\pi(s,a) + (1-\beta)\sum_a \pi(a|s)A_\pi(s,a)\\ &\overset{(b)}{=} \beta\mathop{\mathbb{E}}_{a\sim\tilde{\pi}}[A_\pi(s,a)] - (1-\beta)\alpha H(\pi(\cdot|s))\end{aligned}$$

where the equalities (a) and (b) are due to (12) and Lemma 2, respectively. ■

Lemma 4:

$$\eta(\pi_{\text{mix}}) - \eta(\pi) = \sum_{t=0}^{\infty} \gamma^t \left[\mathbb{E}_{\substack{s\sim P(s_t;\pi_{\text{mix}}) \\ a\sim\pi_{\text{mix}}}} [A_\pi(s,a) + \alpha H(\pi_{\text{mix}}(\cdot|s))] \right].$$

Proof: First note that according to the definition of advantage function and soft bellman equation in (2) and (4), we can get $A_\pi(s_t,a_t) = \mathbb{E}_{s_{t+1}\sim P(\cdot|s_t,a_t)}[r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)]$. Therefore

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\tau_0\sim\pi_{\text{mix}}} \left[\sum_{t=0}^{\infty} \gamma^t [A_\pi(s_t,a_t)] \right] \\
&= \mathop{\mathbb{E}}_{\tau_0\sim\pi_{\text{mix}}} \left[\sum_{t=0}^{\infty} \gamma^t \big[r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) \big] \right] \\
&= \mathop{\mathbb{E}}_{\tau_0\sim\pi_{\text{mix}}} \Bigg[\sum_{t=0}^{\infty} \gamma^t r_t + \Big(\gamma V_\pi(s_1) - V_\pi(s_0) \\
&\qquad + \gamma^2 V_\pi(s_2) - \gamma V_\pi(s_1) + \cdots \Big) \Bigg] \\
&= \mathbb{E}_{\tau_0\sim\pi_{\text{mix}}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] - \mathbb{E}_{s_0}[V_\pi(s_0)] \\
&= \mathbb{E}_{\tau_0\sim\pi_{\text{mix}}} \Bigg[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha H(\pi_{\text{mix}}(\cdot|s_t)) \\
&\qquad - \sum_{t=0}^{\infty} \gamma^t [\alpha H(\pi_{\text{mix}}(\cdot|s_t))] \Bigg] - \eta(\pi) \\
&= \eta(\pi_{\text{mix}}) - \eta(\pi) - \mathbb{E}_{\tau_0\sim\pi_{\text{mix}}} \left[\sum_{t=0}^{\infty} \gamma^t [\alpha H(\pi_{\text{mix}}(\cdot|s_t))] \right].
\end{aligned}$$

Also we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\tau_0\sim\pi_{\text{mix}}} \left[\sum_{t=0}^{\infty} \gamma^t [A_\pi(s_t,a_t) + \alpha H(\pi_{\text{mix}}(\cdot|s_t))] \right] \\
&= \sum_{t=0}^{\infty} \gamma^t \left[\mathbb{E}_{\substack{s\sim P(s_t;\pi_{\text{mix}}) \\ a\sim\pi_{\text{mix}}}} [A_\pi(s,a) + \alpha H(\pi_{\text{mix}}(\cdot|s))] \right].
\end{aligned}$$

Hence, we can derive the formula for Lemma 4. ■

Lemma 5: For any given state s, we can decompose the entropy of the mixed policy as

$$\begin{aligned}
H(\pi_{\text{mix}}(\cdot|s)) = D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s)) + \beta H(\tilde{\pi}(\cdot|s)) \\
+ (1-\beta) H(\pi(\cdot|s)).
\end{aligned}$$

Proof: By definition

$$\begin{aligned}
&H(\pi_{\text{mix}}(\cdot|s)) \\
&= -\sum_a \Big\{ \big[(1-\beta)\pi(a|s) + \beta\tilde{\pi}(a|s) \big] \\
&\qquad \cdot \log\big[(1-\beta)\pi(a|s) + \beta\tilde{\pi}(a|s) \big] \Big\} \\
&= \sum_a \beta \left[\tilde{\pi}(a|s) \log \frac{\tilde{\pi}(a|s)}{(1-\beta)\pi(a|s) + \beta\tilde{\pi}(a|s)} \right] \\
&\quad + \sum_a (1-\beta) \left[\pi(a|s) \log \frac{\pi(a|s)}{(1-\beta)\pi(a|s) + \beta\tilde{\pi}(a|s)} \right] \\
&\quad - \sum_a \big[\beta\tilde{\pi}(a|s) \log \tilde{\pi}(a|s) \big] \\
&\quad - \sum_a \big[(1-\beta)\pi(a|s) \log \pi(a|s) \big] \\
&= D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s)) + \beta H(\tilde{\pi}(\cdot|s)) + (1-\beta) H(\pi(\cdot|s)).
\end{aligned}$$

■

Lemma 6:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{s\sim P(s_t;\pi_{\text{mix}}) \\ a\sim\pi_{\text{mix}}}} [A_\pi(s,a) + \alpha H(\pi_{\text{mix}}(\cdot|s))] \\
&\geq \beta \mathop{\mathbb{E}}_{\substack{s\sim P(s_t;\pi) \\ a\sim\tilde{\pi}}} \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] - 2\beta\rho_t\varepsilon \\
&\quad + \alpha \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})} \big[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s)) \big]
\end{aligned}$$

where $\varepsilon = \max_s |\mathbb{E}_{a\sim\tilde{\pi}}[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s))]|$ and $\rho_t = 1 - (1-\beta)^t$.

Proof:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{s\sim P(s_t;\pi_{\text{mix}}) \\ a\sim\pi_{\text{mix}}}} [A_\pi(s,a) + \alpha H(\pi_{\text{mix}}(\cdot|s))] \\
&\overset{(a)}{=} \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})} \Big[\beta \sum_a \tilde{\pi}(a|s) A_\pi(s,a) - (1-\beta)\alpha H(\pi(\cdot|s)) \\
&\qquad + \alpha D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s)) + \beta\alpha H(\tilde{\pi}(\cdot|s)) + (1-\beta)\alpha H(\pi(\cdot|s)) \Big]
\end{aligned}$$

$$\begin{aligned}
&\beta \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] \\
&= \beta(1-\rho_t) \mathop{\mathbb{E}}_{s\sim P(s_t|c_t=0;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] + \beta\rho_t \mathop{\mathbb{E}}_{s\sim P(s_t|c_t\geq 1;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] \\
&= \beta \mathop{\mathbb{E}}_{s\sim P(s_t|c_t=0;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] - \beta\rho_t \mathop{\mathbb{E}}_{s\sim P(s_t|c_t=0;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] \\
&\quad + \beta\rho_t \mathop{\mathbb{E}}_{s\sim P(s_t|c_t\geq 1;\pi_{\text{mix}})} \left[\sum_a \tilde{\pi}(a|s) \big[A_\pi(s,a) + \alpha H(\tilde{\pi}(\cdot|s)) \big] \right] \qquad (22)
\end{aligned}$$

$$= \beta \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] + \alpha \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})}\left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right]$$

where the equality (a) is according to Lemmas 3 and 5, and the mixed policy is taken as a mixture of the policy π and the referential policy $\tilde{\pi}$ received from others. In other words, to sample from π_{mix}, we first draw a Bernoulli random variable, which tells us to choose π with probability $(1-\beta)$ and choose $\tilde{\pi}$ with probability β. Let c_t be the random variable that indicates the number of times $\tilde{\pi}$ was chosen before time t. $P(s_t;\pi)$ is the distribution over states at time t while following π. We can condition on the value of c_t to break the probability distribution into two pieces, with $P(c_t=0)=(1-\beta)^t$, and $\rho_t = P(c_t \geq 1) = 1-(1-\beta)^t$. Thus, we can get (22), shown at the bottom of the previous page. Recalling the definition $\varepsilon = \max_s |\mathbb{E}_{a\sim\tilde{\pi}}[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))]|$, we have

$$\mathop{\mathbb{E}}_{s\sim P(s_t|c_t=0;\pi_{\text{mix}})}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] \leq \varepsilon$$

$$\mathop{\mathbb{E}}_{s\sim P(s_t|c_t\geq 1;\pi_{\text{mix}})}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] \geq -\varepsilon.$$

Therefore, (22) can be reorganized as

$$\begin{aligned}
&\beta \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] \\
&\geq \beta \mathop{\mathbb{E}}_{s\sim P(s_t|c_t=0;\pi_{\text{mix}})}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] - 2\beta\rho_t\varepsilon \\
&= \beta \mathop{\mathbb{E}}_{s\sim P(s_t;\pi)}\left[\sum_a \tilde{\pi}(a|s)\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big]\right] - 2\beta\rho_t\varepsilon
\end{aligned}$$

in which $P(s_t|c_t=0;\pi_{\text{mix}}) = P(s_t;\pi)$. ■

Next, we are ready to prove Theorem 1.

Proof: According to Lemmas 4 and 6, we have

$$\begin{aligned}
&\eta(\pi_{\text{mix}}) - \eta(\pi) \\
&= \sum_{t=0}^{\infty}\gamma^t\left[\mathbb{E}_{\substack{s\sim P(s_t;\pi_{\text{mix}})\\ a\sim\pi_{\text{mix}}}}[A_\pi(s,a)+\alpha H(\pi_{\text{mix}}(\cdot|s))]\right] \\
&\geq \beta\sum_{t=0}^{\infty}\gamma^t \mathop{\mathbb{E}}_{\substack{s\sim P(s_t;\pi)\\ a\sim\tilde{\pi}}}\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big] - 2\beta\varepsilon\sum_{t=0}^{\infty}\gamma^t\rho_t \\
&\quad + \alpha\sum_{t=0}^{\infty}\gamma^t \mathop{\mathbb{E}}_{s\sim P(s_t;\pi_{\text{mix}})}\left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right] \\
&= \beta \mathop{\mathbb{E}}_{\substack{s\sim d_\pi\\ a\sim\tilde{\pi}}}\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big] - 2\beta\varepsilon\sum_{t=0}\gamma^t\big[1-(1-\beta)^t\big] \\
&\quad + \alpha \mathop{\mathbb{E}}_{s\sim d_{\pi_{\text{mix}}}}\left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right] \\
&= \beta \mathop{\mathbb{E}}_{\substack{s\sim d_\pi\\ a\sim\tilde{\pi}}}\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big] - \frac{2\gamma\varepsilon\beta^2}{(1-\gamma)(1-\gamma(1-\beta))} \\
&\quad + \alpha \mathop{\mathbb{E}}_{s\sim d_{\pi_{\text{mix}}}}\left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right] \\
&\geq \beta \mathop{\mathbb{E}}_{\substack{s\sim d_\pi\\ a\sim\tilde{\pi}}}\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big] - \frac{2\gamma\varepsilon\beta^2}{(1-\gamma)^2} \\
&\quad + \alpha \mathop{\mathbb{E}}_{s\sim d_{\pi_{\text{mix}}}}\left[D_{\text{JS}}^{\beta}(\tilde{\pi}(\cdot|s)\|\pi(\cdot|s))\right].
\end{aligned}$$

We have the theorem. ■

C. Derivation of (19)

Lemma 7:

$$\mathbb{A}_\pi^+(\tilde{\pi}) \approx \mathbb{E}_{s_t,a_t\sim\mathcal{D}}\left[\left(\frac{\tilde{\pi}_{\tilde{\theta}}(a_t|s_t)-\pi_\theta(a_t|s_t)}{\pi_t(a_t|s_t)}\right)\min_{x\in 1,2} Q_{\omega_x}(s_t,a_t) + \alpha\big[H\big(\tilde{\pi}_{\tilde{\theta}}(\cdot|s_t)\big) - H(\pi_\theta(\cdot|s_t))\big]\right].$$

Proof: Following the definition of policy advantage in (14):

$$\begin{aligned}
\mathbb{A}_\pi^+(\tilde{\pi}) &= \mathop{\mathbb{E}}_{s\sim d_\pi, a\sim\tilde{\pi}}\big[A_\pi(s,a)+\alpha H(\tilde{\pi}(\cdot|s))\big] \\
&\overset{(a)}{=} \mathop{\mathbb{E}}_{s\sim d_\pi}\left[\sum_a \tilde{\pi}(a|s)Q^\pi(s,a) - V^\pi(s) + \alpha H(\tilde{\pi}(\cdot|s))\right] \\
&\overset{(b)}{=} \mathop{\mathbb{E}}_{s\sim d_\pi}\left[\sum_a \tilde{\pi}(a|s)Q^\pi(s,a) - \sum_a \pi(a|s)Q^\pi(s,a) - \alpha H(\pi(\cdot|s)) + \alpha H(\tilde{\pi}(\cdot|s))\right]
\end{aligned}$$

where the equality (a) is according to the state-action advantage value in (2) while (b) follows from (4). Finally, by using Monte Carlo and importance sampling, we can get the lemma. ■

REFERENCES

[1] X. Yu et al., "Communication-efficient cooperative multi-agent PPO via regulated segment mixture in Internet of Vehicles," in *Proc. IEEE Globecom*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 3003–3008.

[2] Y. Du, J. Chen, C. Zhao, C. Liu, F. Liao, and C.-Y. Chan, "Comfortable and energy-efficient speed control of autonomous vehicles on rough pavements using deep reinforcement learning," *Transp. Res. Part C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103489.

[3] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Trans. Intell. Veh.*, vol. 6, no. 2, pp. 221–231, Jun. 2020.

[4] T. Li et al., "Applications of multi-agent reinforcement learning in future Internet: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1240–1279, 2nd Quart., 2022.

[5] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[6] 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.

[7] H. Shi, D. Chen, N. Zheng, X. Wang, Y. Zhou, and B. Ran, "A deep reinforcement learning based distributed control strategy for connected automated vehicles in mixed traffic platoon," *Transp. Res. Part C, Emerg. Technol.*, vol. 148, Mar. 2023, Art. no. 104019.

[8] Z. He, L. Dong, C. Song, and C. Sun, "Multiagent soft actor–critic based hybrid motion planner for mobile robots," *IEEE Trans. Neural Netw. Learn. Sys.*, vol. 34, no. 12, pp. 10980–10992, Dec. 2022.

[9] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor–critic for mixed cooperative-competitive environments," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 6382–6393.

[10] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, Nov. 2022, pp. 1–14.

[11] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 2974–2982.

[12] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 2145–2153.

[13] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated reinforcement learning: Techniques, applications, and open challenges," 2021, *arXiv:2108.11887*.

[14] X. Xu, R. Li, Z. Zhao, and H. Zhang, "The gradient convergence bound of federated multi-agent reinforcement learning with efficient communication," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 507–528, Jan. 2024.

[15] Z. Xie and S. Song, "FedKL: Tackling data heterogeneity in federated reinforcement learning by penalizing kl divergence," *IEEE J. Sel. Areas. Commun.*, vol. 41, no. 4, pp. 1227–1242, Apr. 2023.

[16] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A selective federated reinforcement learning strategy for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1655–1668, Feb. 2023.

[17] S. Han et al., "A multi-agent reinforcement learning approach for safe and efficient behavior planning of connected autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3654–3670, May 2023.

[18] J. Z. Leibo et al., "Scalable evaluation of multi-agent reinforcement learning with melting pot," in *Proc. ICML*, Jul. 2021, pp. 1–13.

[19] A. Taik, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25371–25383, Dec. 2022.

[20] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 119–137, Jan. 2024.

[21] M. Movahedian, M. Dolati, and M. Ghaderi, "Adaptive model aggregation for decentralized federated learning in vehicular networks," in *Proc. Int. Conf. Netw. Serv. Manag. (CNSM)*, Niagara Falls, ON, Canada, Oct. 2023, pp. 1–9.

[22] Z. Joel. "Melting pot: An evaluation suite for multi-agent reinforcement learning." Jul. 2021. Accessed: Jul. 3, 2024. [Online]. Available: https://deepmind.google/discover/blog/melting-pot-an-evaluation-suite-for-multi-agent-reinforcement-learning/

[23] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020.

[24] L. Pacheco, T. Braun, D. Rosário, and E. Cerqueira, "An efficient layer selection algorithm for partial federated learning," in *Proc. Pervasive Comput. Commun. Workshops Affil. Events (PerCom Workshops)*, Biarritz, France, Mar. 2024, pp. 172–177.

[25] L. Barbieri, S. Savazzi, and M. Nicoli, "Communication-efficient distributed learning in V2X networks: Parameter selection and quantization," in *Proc. Globecom*, Rio de Janeiro, Brazil, Dec. 2022, pp. 603–608.

[26] A. Nguyen et al., "Deep federated learning for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, Aachen, Germany, Jun. 2022, pp. 1824–1830.

[27] D. Chen, K. Zhang, Y. Wang, X. Yin, Z. Li, and D. Filev, "Communication-efficient decentralized multi-agent reinforcement learning for cooperative adaptive cruise control," *IEEE Trans. Intell. Veh.*, early access, Feb. 21, 2024.

[28] K. Qu, W. Zhuang, Q. Ye, W. Wu, and X. Shen, "Model-assisted learning for adaptive cooperative perception of connected autonomous vehicles," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8820–8835, Aug. 2024.

[29] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *J. Mach. Learn. Res.*, vol. 22, pp. 9709–9758, Sep. 2021.

[30] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 131–143, Feb. 2022.

[31] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4289–4301, Apr. 2023.

[32] P. Watcharapichat, V. L. Morales, R. C. Fernandez, and P. Pietzuch, "Ako: Decentralized deep learning with partial gradient exchange," in *Proc. ACM Symp. Cloud Comput.*, Santa Clara, CA, USA, Oct. 2016, pp. 84–97.

[33] L. Barbieri, S. Savazzi, and M. Nicoli, "A layer selection optimizer for communication-efficient decentralized federated deep learning," *IEEE Access*, vol. 11, pp. 22155–22173, 2023.

[34] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," 2019, *arXiv:1908.07782*.

[35] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, pp. 1–210, Jun. 2021.

[36] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proc. ICML*, Sydney, NSW, Australia, Aug. 2017, pp. 1352–1361.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. ICML*, Stockholm, Sweden, Jul. 2018, pp. 1–10.

[38] T. Haarnoja et al., "Soft actor–critic algorithms and applications," 2018, *arXiv:1812.05905*.

[39] X. Xu, R. Li, Z. Zhao, and H. Zhang, "Trustable policy collaboration scheme for multi-agent stigmergic reinforcement learning," *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 823–827, Apr. 2022.

[40] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proc. ICML*, Sydney, NSW, Australia, Jul. 2002, pp. 267–274.

[41] J. Kuba et al., "Trust region policy optimization in multi-agent reinforcement learning," in *Proc. ICLR*, Apr. 2022, pp. 1–27.

[42] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 1–16.

[43] J. Schneider, W.-K. Wong, A. Moore, and M. Riedmiller, "Distributed value functions," in *Proc. ICML*, Jun. 1999, pp. 1–8.

[44] E. Ferreira and P. Khosla, "Multi agent collaboration using distributed value functions," in *Proc. IEEE Intell. Veh. Symp.*, Dearbon, MI, USA, Oct. 2000, pp. 1–6.

[45] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 8667–8681, Oct. 2017.

[46] Z. Xia, J. Du, and Y. Ren, "Convergence theory of generalized distributed subgradient method with random quantization," 2022, *arXiv:2207.10969*.

[47] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE J. Sel. Areas. Commun.*, vol. 39, no. 12, pp. 3723–3741, Dec. 2021.

[48] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems," *Knowl. Eng. Rev.*, vol. 27, pp. 1–31, Mar. 2012.

[49] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. ICML*, Jun. 1993, pp. 330-337.

[50] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," 2020, *arXiv:2006.07869*.

[51] P. Dhariwal et al. "OpenAI baselines." 2017. [Online]. Available: https://github.com/openai/baselines

[52] C. S. de Witt et al., "Is independent learning all you need in the starcraft multi-agent challenge?" 2020, *arXiv:2011.09533*.

[53] G. Sartoretti, Y. Wu, W. Paivine, T. S. Kumar, S. Koenig, and H. Choset, *Distributed Reinforcement Learning for Multi-Robot Decentralized Collective Construction* (Springer Proceedings in Advanced Robotics). Boulder, CO, USA: Springer, 2019. [Online]. Available: https://dblp.org/db/conf/dars/dars2018.html

[54] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. ICML*, New York, NY, USA, Jun. 2016, pp. 1928–1937.

[55] X. Xu, R. Li, Z. Zhao, and H. Zhang, "Stigmergic independent reinforcement learning for multiagent collaboration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4285–4299, Sep. 2022.

[56] M. Camelo, M. Claeys, and S. Latré, "Parallel reinforcement learning with minimal communication overhead for IoT environments," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1387–1400, Feb. 2020.

[57] Y. Gao, L. Zhang, L. Wang, K.-K. R. Choo, and R. Zhang, "Privacy-preserving and reliable decentralized federated learning," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2879–2891, Jul. 2023.

[58] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. AISTATS*, Aug. 2020, pp. 1–10.

[59] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2018, pp. 1–11.

[60] Z. Tang, S. Shi, B. Li, and X. Chu, "GossipFL: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 3, pp. 909–922, Mar. 2023.
[61] X. Gao, Y. Sun, H. Chen, X. Xu, and S. Cui, "Joint computing, pushing, and caching optimization for mobile edge computing networks via soft actor–critic learning," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9269–9281, Mar. 2023.
[62] J. Duan, Y. Guan, S. E. Li, Y. Ren, Q. Sun, and B. Cheng, "Distributional soft actor–critic: Off-policy reinforcement learning for addressing value estimation errors," *IEEE Trans. Neural Netw. Learn. Sys.*, vol. 33, no. 11, pp. 6584–6598, Nov. 2022.
[63] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor–critic methods," in *Proc. ICML*, Stockholm, Sweden, Jul. 2018, pp. 1–10.
[64] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 4431–4506, 2021.
[65] F. Nielsen, "On the Jensen–Shannon symmetrization of distances relying on abstract means," *Entropy*, vol. 21, p. 485, May 2019.
[66] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Proc. 19th IFIP WG 6.1 Int. Conf. Distrib. Appl. Interoper. Syst.*, Lyngby, Denmark, Jun. 2019, pp. 74–90.
[67] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6G connected vehicles," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100396.
[68] S. M. Kakade, "A natural policy gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, Dec. 2001, pp. 1–8.
[69] "Vehicular communications; basic set of applications; analysis of the collective perception service (CPS); (Release 2), Version 2.1.1," ETSI, Sophia Antipolis, France, Rep. 103 562, Dec. 2021.
[70] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, pp. 251–276, Feb. 1998.
[71] T. George, C. Laurent, X. Bouthillier, N. Ballas, and P. Vincent, "Fast approximate natural gradient descent in a Kronecker factored eigenbasis," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 9573–9583.
[72] J. Martens et al., "Deep learning via Hessian-free optimization," in *Proc. ICML*, Haifa, Israel, Jun. 2010, pp. 735–742.
[73] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1270–1286, Apr. 2022.
[74] E. Vinitsky et al., "Benchmarks for reinforcement learning in mixed-autonomy traffic," in *Proc. 2nd Conf. Robot Learn.*, Zürich, Switzerland, Oct. 2018, pp. 399–409.
[75] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, p. 1805, Aug. 2000.
[76] M. Yang et al., "Dynamic V2V channel measurement and modeling at street intersection scenarios," *IEEE Trans. Antennas. Propag.*, vol. 71, no. 5, pp. 4417–4432, May 2023.
[77] J. Thota, N. F. Abdullah, A. Doufexi, and S. Armour, "V2V for vehicular safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2571–2585, Jun. 2020.
[78] "Service requirements for enhanced V2X scenarios, (Release 18), Version 18.0.1," 3GPP, Sophia Antipolis, France, Rep. 22.186, 2024.
[79] B. Xiao et al., "Stochastic graph neural network-based value decomposition for multi-agent reinforcement learning in urban traffic control," in *Proc. IEEE 97th Veh. Technol. Conf.*, Florence, Italy, Jun. 2023, pp. 1–7.

Xiaoxue Yu (Student Member, IEEE) received the B.E. degree in communication engineering from Xidian University, Xi'an, China. She is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. Her research interests include communications in distributed learning and multiagent reinforcement learning.

Rongpeng Li (Senior Member, IEEE) received the B.E. degree from Zhejiang University, Hangzhou, China, in June 2015, and the Ph.D. degree from Xidian University, Xi'an, China, in June 2010. From August 2015 to September 2016, he was a Research Engineer with the Wireless Communication Laboratory, Huawei Technologies Company Ltd., Shanghai, China. He was a Visiting Scholar with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K., from February 2020 to August 2020. He is currently an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His research interest currently focuses on networked intelligence for communications evolving.

Chengchao Liang received the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, ON, Canada, in 2017, awarded the Senate Medal. He is currently a Full Professor with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include wireless communications, satellite networks, Internet protocols, and optimization theory.

Zhifeng Zhao (Member, IEEE) received the B.E. degree in computer science, the M.E. degree in communication and information systems, and the Ph.D. degree in communication and information systems from the PLA University of Science and Technology, Nanjing, China, in 1996, 1999, and 2002, respectively. From 2002 to 2004, he was as a Postdoctoral Researcher with Zhejiang University, Hangzhou, China. From 2005 to 2006, he was as a Senior Researcher with the PLA University of Science and Technology. He was an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University from 2006 to 2019. He is currently with Zhejiang Lab, Hangzhou, as the Chief Engineering Officer, as well as Zhejiang University as an Adjunct Professor. His research areas include software-defined networks, wireless networks in 6G, computing networks, and collective intelligence.