Multi-Agent Reinforcement Learning for Large Model-Aided Mobile Applications

Yiwen Zhan
Department of Informatics and Communication Engineering, Xiamen University.

Abstract—**Multi-agent reinforcement learning utilizes the observations and learning experiences shared among the agents to accelerate optimization efficiency under partial observations, but the performance degrades due to the excessive communications and large volumes of irrelevant experiences. In this paper, we propose a multi-agent reinforcement learning scheme to optimize communication decision, the cooperative agent selection, and the required number of sharing experiences based on the task features and previous contributions of neighbors extracted by large model to improve the quality of service for mobile applications in wireless networks. The experiences with high temporal difference error are shared and utilized based on the attention mechanism, which extracts the correlation with the reachable neighbors set to improve optimization efficiency. As a case study, the proposed communication scheme is implemented in the vision transformer-aided collaborative perception system based on connected autonomous vehicles and roadside unit to support 3D object detection, and the performance gain is verified via simulation results.**

Index Terms—**Multi-agent reinforcement learning, large model, communications, mobile applications, wireless networks.**

I. Introduction

Multi-agent reinforcement learning (MARL) in wireless communications enabled learning agents such as mobile devices to optimize task policies such as the radio resource allocation to support reliable and low-latency applications and services such as augmented reality [1]. However, most learning agents only have partial observations on the environment such as the radio channel states and suffer from non-stationarity caused by the changing policies of other agents, which in turn degrades the learning performance and the quality of service (QoS) in dynamic networks.

As the neighboring learning agents share a common environment and can exchange observations via wireless networks, MARL algorithms can exploit the shared environment information in the state formulation to enhance the agent receptive field and improve the policy optimization efficiency [2], [3]. For example, the multi-agent deep Q-network-based unmanned aerial vehicle communication scheme ECOM exploits the shared observations such as the signal-to-interference-plus-noise ratio and task similarity to reduce the transmission delay and packet loss rate against smart jamming. However, the performance of ECOM is hindered by long communication latency and data redundancy due to the excessive communications in limited radio bandwidth.

Moreover, MARL agents share learning experiences and parameters such as Q-values and network weights to accelerate learning [4], [5]. For example, the selective multi-agent prioritized experience relay scheme SUPER enable each agent select whether to share experiences by comparing absolute temporal difference (TD) error with deterministic quantile, which are inserted into all the other agents' replay buffers [5]. However, SUPER suffers from performance degradation in terms of convergence speed and the application QoS in dynamic wireless networks due to the pre-determined sharing quantity and the exchange of irrelevant experiences.

In this paper, we propose a multi-agent communication scheme to optimize the communication decision, the cooperative agent selection and the number of shared experiences based on the radio channel states, previous communication overhead, and the task features and contributions of neighbors extracted by transformer. The task-relevant observations are exchanged to reduce redundant communications under limited bandwidth. The experiences with high TD error are shared, and The attention mechanism compared the reachable neighbors set with the selected cooperative agents to calculate the importance weight of the sampled experiences to update the network weights to accelerate convergence and improve application QoS in dynamic wireless networks.

The rest of this paper is organized as follows. The related work is reviewed in Section II, and the system model is given in Section III. we present the multi-agent communication scheme in Sections IV. The case study on the anti-jamming collaborative perception is provided in Section V, followed by the simulation results in Sections VI. Finally, the conclusion is summarized in Section VII.

II. Related Work

MARL such as multi-agent deep deterministic policy gradient (DDPG) enables multi-agent systems in wireless networks to optimize transmission policies such as resource allocation and trajectory design [6]–[9]. For example, the multi-connectivity ultra reliable low latency communication system for vehicular networks in [6] applies transformer associated proximal policy optimization algorithm to optimize power allocation and thus minimizes inter-cell interference and energy consumption.The unmanned aerial vehicle empowered aerial computing system in [7] employs multi-agent cooperative Q-learning and multi-agent cooperative actor-critic algorithm to optimize trajectory planning, as well as the placement of both the departure station and hovering points, aiming to maximize energy efficiency and service fairness, respectively.

Learning information exchange including environment observations and learning experiences among learning agents enhances the RL performance under partial observation [3]–[5]. For example, the multi agent wireless system in [10] shares information with other agents via neural network parameters and analyzes the effect of the parameter sharing frequency on the convergence speed. The selective multi-agent prioritized experience relay in [5] applies deterministic quantile to selects and shares the experiences with the largest absolute TD errors in the top β-quantile to improve learning efficiency.

Efficient communication is promising for MARL to enhance learning performance with attention experience sharing, and as examples, RIAL and DIAL in [11] and CommNet in [12] learn to communicate between agents in cooperative environments with partial observations. For example, the communication protocol in [11] uses deep Q-learning with recurrent networks to encode past observations, actions and local observations into binary messages and thus reduces communication overhead but has information loss in large-scale wireless networks. The communication information mainly consists of original observations, encoded observations [13], [14] and intended actions [15]. In particular, an intention sharing scheme in [15] applies the attention mechanism to generate an imagined trajectory based on the shared messages from other agents and enhance the coordination of multi-agent systems.

III. System Model

Each learning agent has N neighboring learning agents $\mathcal{N} = \{1, ..., N\}$ to exchange learning information including observations and experiences based on the carrier sense multiple access/collision avoidance access (CSMA/CA) [16] and optimize task policies such as channel selection and power allocation to support transformer-based mobile applications such as object detection and tracking, semantic segmentation, and scene understanding.

At time slot k, the learning agent observes local observation $\boldsymbol{o}^{(k)}$, estimates the channel states $\boldsymbol{h}^{(k)} = \left[h_i^{(k)}\right]_{i\in\mathcal{N}}$, receives task features $\boldsymbol{\nu}^{(k)} = \left[\nu_i^{(k)}\right]_{i\in\mathcal{N}}$ and previous contributions of neighbors $\boldsymbol{\varrho}^{(k)} = \left[\varrho_{1\leq i\leq N}^{(k)}\right] \subset [0,1]^N$ extracted by transformer in central server, and chooses cooperative agents $\boldsymbol{x}_1^{(k)} \subset \mathcal{N}$ and requests shared observations and at most B experiences $x_2 \in \{1, ..., B\}$. If $\boldsymbol{x}_1^{(k)} = \varnothing$, the learning agent selects task action solely based on local observations. The request message $\mathcal{T}^{(k)}$ consisting of the selected agent ID and the required number of experiences $x_2^{(k)}$ is broadcast to the selected agents $\boldsymbol{x}_1^{(k)}$.

Upon receiving the learning request message, the cooperative agent $i \in \boldsymbol{x}_1^{(k)}$ obtains local observation $\tilde{\boldsymbol{o}}_i^{(k)}$ and top $x_2^{(k)}$ experiences with largest TD error $\boldsymbol{\varepsilon}_i^{(k)}$, which are sent back to learning agent within the response message $\mathcal{R}_i^{(k)} = \left\{i, \tilde{\boldsymbol{o}}_i^{(k)}, \boldsymbol{\varepsilon}_i^{(k)}\right\}$. The shared experiences $\{\boldsymbol{\varepsilon}_i^{(k)} | i \in \boldsymbol{x}_1^{(k)}\}$ are stored in the communication memory pool $\mathcal{B}$ for the updating of network weights. The shared observations $\{\tilde{\boldsymbol{o}}_i^{(k)} | i \in \boldsymbol{x}^{(k)}\}$ are used to formulate the task learning state $\boldsymbol{s}^{(k)}$, and choose the task action $\boldsymbol{a}^{(k)} \in \mathrm{A}$. Based on the task performance $\boldsymbol{\rho}^{(k)}$ such as perception accuracy, the learning agent evaluates the task reward $r^{(k)}$ and communication cost, i.e., the number of selected cooperative agents $\xi^{(k)}$ to guide the communication policy selection. Time slot k is omitted if no confusion occurs in the subsequent sections.

IV. Multi-Agent Communication Scheme

We propose a multi-agent communication scheme named MACS for transformer-aided mobile applications in wireless networks that enables learning agents to optimize communication decision, cooperative agents, and the required number of sharing experiences for faster learning under partial observation. Experiences with high TD error are shared and utilized based on the attention mechanism, which measures the correlation with the reachable neighbor set to improve the optimization efficiency of communication policies.

The multi-agent reinforcement learning scheme MACS jointly chooses the communication action $\boldsymbol{x} = [\boldsymbol{x}_1, x_2]$ including cooperative agents $\boldsymbol{x}_1 \subset \mathcal{N} = \{1, ..., N\}$ and the number of experience $x_2 \in \{1, 2, \ldots, B\}$, as well as task action $\boldsymbol{a}$ to maximize the discounted long-term reward. The communication state $\boldsymbol{s}^{(k)}$ consists of local observations $\boldsymbol{o}$, channel states $\boldsymbol{h}$, task features $\boldsymbol{\nu}$ and previous contribution $\boldsymbol{\varrho}$ received from central server, and previous communication cost ξ, i.e.,

$$\boldsymbol{s}^{(k)} = [\boldsymbol{o}, \boldsymbol{h}, \boldsymbol{\nu}, \boldsymbol{\varrho}, \xi] \in \mathbf{S}. \tag{1}$$

The learning agent chooses the communication action $\boldsymbol{x}$ from action set $\mathcal{A} = \{1, ..., N\} \times \{1, 2, \ldots, B\}$ according to the policy distribution $\pi(\boldsymbol{s}, \boldsymbol{x})$, i.e.,

$$\pi(\boldsymbol{s}, \boldsymbol{x}) = \frac{\exp\left(Q(\boldsymbol{s}, \boldsymbol{x})\right)}{\sum_{\bar{\boldsymbol{x}}\in\mathcal{A}} \exp\left(Q(\boldsymbol{s}, \bar{\boldsymbol{x}}) + 1\right)}, \tag{2}$$

which depends on the Q-values $Q\left(\boldsymbol{s}^{(k)}, \cdot; \boldsymbol{\omega}\right)$ estimated by Q-network and the communication range R.

The request message $\mathcal{T} = \{\boldsymbol{o}, x_2\}$ is sent to selected agents $i \in \boldsymbol{x}_1$ for the shared messages. Upon receiving the learning response message $\mathcal{R}_i = \{i, \tilde{\boldsymbol{o}}_i, \boldsymbol{\varepsilon}_i\}$, the experiences $\boldsymbol{\varepsilon}_i$ are stored in the communication memory pool $\mathcal{B}$ for the update of network weights, and the shared observation $\tilde{\boldsymbol{o}}_i$ is used to formulate the task state $\tilde{\boldsymbol{s}}^{(k)} = [\boldsymbol{o}, \tilde{\boldsymbol{o}}_{i\in\boldsymbol{x}_1}] \in \tilde{\mathbf{S}}$. The learning agent chooses the task action $\boldsymbol{a}$ from action set $\mathbf{A}$ based on the V-values $V\left(\tilde{\boldsymbol{s}}^{(k)}, \cdot; \boldsymbol{\theta}\right)$ output from V-network according to the $episilon$-greedy method.

The learning agent measures the task performances $\boldsymbol{\rho}$ such as perception accuracy to evaluate the task reward r for updating the network weights $\boldsymbol{\theta}$ with the experience replay technique. The task learning experience $\{\tilde{\boldsymbol{s}}^{(k)}, \boldsymbol{a}^{(k)}, r^{(k)}, \tilde{\boldsymbol{s}}^{(k+1)}\}$ is formulated and stored in the task memory pool $\mathcal{D}$, and G experiences are uniformly sampled to update the network weights $\boldsymbol{\theta}$ via stochastic gradient descent algorithm such as Adam, given as

$$\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}'} \frac{1}{G} \sum_{j=1}^{G} \left(r^{(j)} + \gamma \max_{\hat{\boldsymbol{a}} \in \mathbf{A}} V\left(\tilde{\boldsymbol{s}}^{(j+1)}, \hat{\boldsymbol{a}}; \boldsymbol{\theta}'\right) - V\left(\tilde{\boldsymbol{s}}^{(j)}, \boldsymbol{a}^{(j)}; \boldsymbol{\theta}\right) \right)^2, \tag{3}$$

where γ represents the importance of the future reward in the learning process. The communication reward $\tilde{r}$ is evaluated based on the task reward r and the communication cost ξ, i.e., $\tilde{r} = r - \xi$. The communication experience $\left\{\boldsymbol{s}^{(k)}, \boldsymbol{x}_1^{(k)}, x_2^{(k)}, \tilde{r}^{(k)}, \boldsymbol{s}^{(k+1)}\right\}$ is formulated and stored in the communication memory pool $\mathcal{B}$, and G experiences are uniformly sampled, i.e.,

$$\mathcal{F} = \left\{\boldsymbol{s}^{(j)}, \boldsymbol{x}_1^{(j)}, x_2^{(j)}, r^{(j)}, \boldsymbol{s}'^{(j)}\right\}_{j=1}^{G} \tag{4}$$

to update network weights $\boldsymbol{\omega}$. The scaled dot-product attention mechanism is applied to measure the importance weight $\boldsymbol{\eta} = \left\{\eta^{(j)} \mid 1 \leq j \leq G\right\}$ of sampled experiences, which takes the reachable neighbor set $\mathcal{C}^{(k)} = \{i \mid d_{i \in \mathcal{N}} \leq R\}$ within the communication range R and the cooperative agent set $\boldsymbol{x}_1^{(j)}$ as input. More specially, the reachable neighbor set $\mathcal{C}^{(k)}$ is transformed to query with $\boldsymbol{W}^{\mathrm{Q}} \in \mathbb{R}^{N \times V}$, and the cooperative agent set $\boldsymbol{x}_1^{(j)}$ is transformed to key and value with $\boldsymbol{W}^{\mathrm{K}}$ and $\boldsymbol{W}^{\mathrm{V}} \in \mathbb{R}^{N \times V}$, respectively. The learning agent uses the weighted sum of value and the dot product and softmax function to calculates the importance weight for sampled experience $\boldsymbol{\eta}$, and update network weights $\boldsymbol{\omega}$ by minimizing the loss function given by

$$\mathcal{L}(\boldsymbol{\omega}) = \sum_{j=1}^{\mathrm{G}} \eta^{(j)} \left(\tilde{r}^{(j)} + \gamma \max_{\bar{\boldsymbol{x}} \in \mathcal{A}} Q\left(\boldsymbol{s}^{(j+1)}, \bar{\boldsymbol{x}}; \boldsymbol{\omega}'\right) - Q\left(\boldsymbol{s}^{(j)}, \boldsymbol{x}^{(j)}; \boldsymbol{\omega}\right) \right)^2. \tag{5}$$

V. CASE STUDY: MARL-BASED ANTI-JAMMING COLLABORATIVE PERCEPTION

As a case study, the proposed multi-agent communication scheme is implemented in the MARL-based anti-jamming collaborative perception to improve the task performance such as perception accuracy and speed. More specifically, each connected autonomous vehicle (CAV) as the learning agent is assumed to have N neighboring CAVs, which chooses the offloading region of request map $\boldsymbol{\kappa}^{(k)}$, transmit channel $c^{(k)} \in \{1, 2, \ldots, F\}$ and power $p^{(k)} \in [P_{\min}, P_{\max}]$ to support vision transformer-aided mobile applications such as smart transportation. A smart jammer applies the signal detection techniques to sense the ongoing transmission and chooses the jamming channel and power to degrade the perception performance.

At time slot k, RSU divides the feature map into L regions each with m-bits, calculates the spatial confidence score $\boldsymbol{\varpi} = [\varpi_i]_{1 \leq i \leq L}$ based on the received request map from RSU, measures the received jamming strength σ, and formulates the local observation $\boldsymbol{o}$ with previous perception latency τ, i.e.,

$$\boldsymbol{o}^{(k)} = [m, \boldsymbol{\varpi}, \sigma, \tau]. \tag{6}$$

Based on the local observation $\boldsymbol{o}^{(k)}$, the channel state $\boldsymbol{h}$ with neighboring CAVs, previous contributions of neighbors $\boldsymbol{\varrho}^{(k)}$ and task features such as vehicle density extracted by vision transformer in RSU $\boldsymbol{\nu}^{(k)}$, and the communication cost ξ, the communication learning state $\tilde{\boldsymbol{s}}^{(k)}$ is formulated as

$$\tilde{\boldsymbol{s}}^{(k)} = [m, \boldsymbol{\varpi}, \sigma, \tau, \boldsymbol{h}, \boldsymbol{\varrho}, \xi]. \tag{7}$$

With the communication state $\tilde{\boldsymbol{s}}^{(k)}$ as input, the Q-network estimates the communication Q-value $Q\left(\tilde{\boldsymbol{s}}^{(k)}, \cdot; \boldsymbol{\omega}\right)$ to choose the cooperative CAVs $\boldsymbol{x}_1$ and the number of shared experience x_2 via 2. Based on the shared observations $\tilde{\boldsymbol{o}}^{(k)}$, CAV formulates the task state $\boldsymbol{s}^{(k)} = [\boldsymbol{o}^{(k)}, \tilde{\boldsymbol{o}}^{(k)}]$, and chooses the anti-jamming perception policy including the offloading region of feature maps $\boldsymbol{\kappa}^{(k)}$, the transmit channel $c^{(k)}$, and power $p^{(k)}$ based on the V-values $V\left(\boldsymbol{s}^{(k)}, \cdot; \boldsymbol{\theta}\right)$ according to the *epsilon*-greedy method.

After receiving the feature maps, the RSU performs the feature fusion model to obtain the detection result $\mathcal{Z}$, and further evaluates the perception accuracy $\rho^{(k)}$ and latency $\tau^{(k)}$, which are sent back to each CAV. The task reward $r^{(k)}$ is evaluated via $r^{(k)} = \rho^{(k)} - \alpha_1 \tau^{(k)}$, where α_1 represent the importance of perception latency. Based on the communication cost $\xi^{(k)}$, the communication reward $\tilde{r}^{(k)}$ is evaluated as $\tilde{r}^{(k)} = r^{(k)} - \xi^{(k)}$. The anti-jamming perception experience is stored in the memory pool $\mathcal{D}$, and the shared experience and communication experience are stored in the memory pool $\mathcal{B}$. The experience replay technique is used to update the network weights of Q-network and V-network with the stochastic gradient descent algorithm according to (3) and (5), respectively.

VI. SIMULATION RESULTS

Simulations were performed based on the feature fusion model V2X vision transformers and the V2XSet dataset in [17] containing 11,447 frames (33,081 samples including frames per agent) to evaluate the performances of collaborative perception in terms of perception accuracy, latency and utility for the LiDAR-based 3D object detection. The perception accuracy is evaluated based on the intersection-over-union (IoU) threshold of 0.7 that represents the proportion of the overlapping area of bounding boxes between the prediction and ground truth (e.g., a vehicle is detected if the proportion of the overlapping area is greater than 0.7).

Each CAV partitions the feature map into four regions each with 1 MB, one out of the six radio channels and the transmit power up to 100 mW that is quantified into 10 levels. The smart jammer is away from the RSU by 200 m estimates the RSSI of vehicular uplink channels, and applies the Q-learning algorithm to choose power up to 80 mW. The channel gain between the RSU and CAV $h = h_0 \left(d\right)^{\alpha} X_{\sigma}$ depends on the

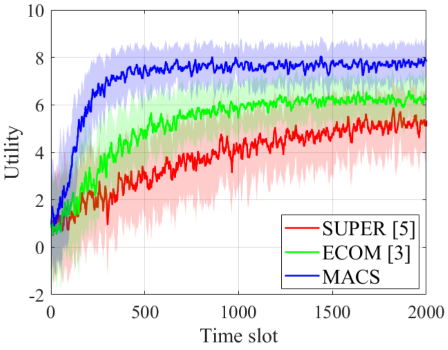

Fig. 1. Performance of the anti-jamming collaborative perception in LiDAR-based 3D object detection.

reference path-loss $h_0 = 68.83$ at reference distance d_0 = 1 m, the path-loss exponent $\alpha = 2.75$, the communication distance d and the shadow fading X_σ that modeled by a zero mean normal distribution with a standard deviation of $\sigma = 5.5$.

The Q-network in the communication policy selection and the R-network in task policy selection are instantiated as the four-layer neural networks including the input and output layer and two hidden layers each with 128 neural nodes. The learning rate of the Q-value update λ = 0.4, the discount factor of reward γ = 0.3, the greedy parameter ϵ decays from 1 to 0.01 after 1000 time slots to make a trade-off between exploration and exploitation, and the number of sampled experiences in the minibatch G = 64.

As shown in Fig. 1, the performance of collaborative perception in vehicular networks averaged by 200 runs each with 2000 time slots is provided. Our proposed multi-agent communication scheme MACS outperforms the benchmarks ECOM and SUPER with 29.5% and 58% higher utility after 1500 time slots, respectively. In addition, MACS converges after about 500 time slots and saves 50% and 66.7% learning time compared with ECOM and SUPER, respectively. The reason is that the proposed scheme enables agents to make necessary communication decisions and share the most relevant observations and experiences, thereby improving anti-jamming collaborative perception efficiency and accelerating learning speed.

VII. Conclusion

In this paper, we proposed multi-agent reinforcement learning scheme for the transformer-aided mobile applications to choose whether to communicate, the cooperative agents and the required number of experience. The task features and previous contributions of neighbors extracted by transformer are exploited in the state formulation to ensure necessary communications and retrieve relevant messages. Shared experiences with high TD error are used to update network weights based on the importance weight computed by the attention mechanism to accelerate learning speed. A a case study, the proposed scheme was implemented in the collaborative perception systems to choose the offloading region of feature maps, transmit channel and power against smart jamming. The simulation results based on five CAVs and an RSU show the performance gains over the benchmark SUPER and ECOM.

References

[1] T. Li, K. Zhu, N. C. Luong, *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, Mar. 2022.

[2] Z. Ding, T. Huang, and Z. Lu, "Learning individually inferred communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* Vancouver, Canada, pp. 22069–22079, 2020.

[3] Z. Lv, L. Xiao, Y. Du, Y. Zhu, S. Han, and Y.-J. Liu, "Efficient communications in multi-agent reinforcement learning for mobile applications," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 12440–12454, 2024.

[4] F. Christianos, L. Schäfer, and S. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* Vancouver, Canada, vol. 33, pp. 10707–10717, 2020.

[5] M. Gerstgrasser, T. Danino, and S. Keren, "Selectively sharing experiences improves multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* New Orleans, Louisiana, vol. 36, pp. 59543–59565, 2023.

[6] J. Xue, K. Yu, T. Zhang, H. Zhou, L. Zhao, and X. Shen, "Cooperative deep reinforcement learning enabled power allocation for packet duplication urllc in multi-connectivity vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8143–8157, Aug. 2024.

[7] M. Tao, X. Li, J. Feng, *et al.*, "Multi-agent cooperation for computing power scheduling in uavs empowered aerial computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 12, pp. 3521–3535, Dec. 2024.

[8] Z. Wei, B. Li, R. Zhang, *et al.*, "Many-to-many task offloading in vehicular fog computing: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2107–2122, Mar. 2024.

[9] J. Ji, L. Cai, K. Zhu, *et al.*, "Decoupled association with rate splitting multiple access in UAV-assisted cellular networks using multi-agent deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2186–2201, Mar. 2024.

[10] F. Hu, Y. Deng, and A. Hamid Aghvami, "Scalable multi-agent reinforcement learning for dynamic coordinated multipoint clustering," *IEEE Transactions on Communications*, vol. 71, no. 1, pp. 101–114, 2023.

[11] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* Barcelona, Spain, p. 2252–2260, 2016.

[12] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* Barcelona, Spain, p. 2252–2260, 2016.

[13] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, "TarMAC: Targeted multi-agent communication," in *Proc. Int. Conf. Mach. Learn. (ICML),* Long Beach, California, vol. 97, pp. 1538–1546, 2019.

[14] T. Lin, M. Huh, C. Stauffer, S.-N. Lim, and P. Isola, "Learning to ground multi-agent communication with autoencoders," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS),* Virtual, pp. 15230–15242, 2021.

[15] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in *Proc.Int. Conf. Learn. Represent. (ICLR),* Vienna, Austria, 2020.

[16] A. Tsertou and D. I. Laurenson, "Revisiting the hidden terminal problem in a CSMA/CA wireless network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 817–831, 2008.

[17] R. Xu, C.-J. Chen, Z. Tu, *et al.*, "V2X-ViTv2: Improved vision transformers for vehicle-to-everything cooperative perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 1, pp. 650–662, Jun. 2025.