SDE−HARL: Scalable Distributed Policy Execution for Heterogeneous-Agent Reinforcement Learning

Toan D. Gian[1], **Mohammad Abdi**[1], **Nathaniel D. Bastian**[2], **Francesco Restuccia**[1]

[1]Northeastern University, United States [2]United States Military Academy, United States
{toan.g, abdi.mo, f.restuccia}@northeastern.edu, nathaniel.bastian@darpa.mil

Abstract

Heterogeneous-agent reinforcement learning (HARL) enables agents to execute cooperative tasks by adopting agent-specific policies. Most of existing HARL methods use individual policy neural networks to ensure monotonic improvement, which leads to substantial computational overhead. The proposed SDE−HARL overcomes this limitation by decomposing each agent's policy neural network into a lightweight *local* neural network and a *global* neural network executed at an edge server. Each local neural network generates and sends a *compressed* latent representation to the edge server, which aggregates the representations and produces agent-specific inferences. As such, SDE−HARL allows to significantly save computing and networking resources while preserving agent-specific behavior. A key feature of SDE−HARL is grouping agents with similar roles via a role-aware mechanism and share partial parameters in their global networks, while an identity-aware mechanism is introduced to promote behavioral diversity among agents within the same group. We prototyped SDE−HARL on an experimental testbed composed of a Jetson Nano and Raspberry PI to measure latency and network resource consumption. We evaluated SDE−HARL's performance on several benchmark datasets, including Google Research Football and StarCraft II. Experimental results show that SDE−HARL reaches up to 90% win rate while reducing latency, energy consumption, and networking overhead respectively by $2\times$, $2.5\times$, and $5\times$ compared to existing work. Source code is available here: https://tinyurl.com/5cvmj259.

Introduction

Motivation. Multi-agent reinforcement learning (MARL) enables cooperative behaviors among intelligent agents operating within complex environments (Gronauer and Diepold 2022; Canese et al. 2021; Li et al. 2022). Such cooperation is essential for solving complex multi-agent decision-making problems, where multiple agents need to coordinate their actions toward achieving shared team rewards (Oroojlooy and Hajinezhad 2022; Ye, Zhang, and Yang 2015; Liu et al. 2021; Yuan et al. 2023). MARL systems are generally categorized into *homogeneous* and *heterogeneous* agent systems. In homogeneous systems, all agents have identical action spaces and policy architectures, making them simpler to implement but often inadequate for tasks requiring specialized agent behaviors. In contrast, HARL (Zhong et al. 2024; Liu et al. 2023a; Kuba et al. 2021) allows agents to independently learn distinct policies, thereby creating a more practical learning framework suitable for real-world deployments.

Existing Limitations. Despite the advantages offered by HARL, two limitations hinder their practical applicability (Du and Ding 2021; Nguyen, Nguyen, and Nahavandi 2020). Existing HARL methods commonly follow a *centralized training decentralized execution (CTDE)* paradigm, deploying distinct policy networks directly on each local agent. This implicitly assumes abundant computational resources are available locally. However, in real-world applications, agents typically are resource-constrained mobile devices, rendering the deployment of large policy networks impractical.

Additionally, previous HARL studies (Kuba et al. 2021; Liu et al. 2023a; Zhong et al. 2024) guarantee monotonic policy improvement by sequentially updating each agent's policy, which inherently requires maintaining separate policy networks per agent. This approach disregards a critical observation: much like human teams, agents often share significant similarities in their decision-making processes. Consequently, existing HARL methods miss opportunities for leveraging shared understanding across agents, resulting in increased training overhead and poor sample efficiency.

Challenges. Recent work (Tahir and Parasuraman 2025) suggests addressing computational constraints by offloading tasks to compute-capable servers. Nonetheless, directly forwarding raw observations to nearby servers leads to a substantial network load, which becomes infeasible at scale. For instance, in (Tahir and Parasuraman 2025), ten agents equipped with LiDAR sensors operating at 10 Hz need to transmit approximately 750 KB of data per policy execution step, highlighting the severity of the networking bottleneck.

Alternatively, sharing policy parameters has emerged as a promising solution in homogeneous systems to enable knowledge sharing, called policy decentralization with shared parameters (PDSP) (Wang et al. 2020a; Iqbal et al. 2021; Wang et al. 2020b). However, directly applying a partial PDSP approach—such as sharing a common policy backbone while using agent-specific heads—to heterogeneous scenarios introduces two significant challenges. Specifically, partial PDSP in HARL settings may: (i) disrupt sequential policy updates, fundamental in existing HARL training methodologies (Kuba et al. 2021); and (ii) fail to encourage behavioral diversity

among agents. Excessive parameter sharing leads to similar agent behaviors under similar observations (Liu et al. 2023b; Bettini, Kortvelesy, and Prorok 2024; Yu et al. 2024), limiting exploration capabilities and effective cooperation.

These challenges raise two fundamental research questions: *(1) How can scalable HARL architectures be designed to operate efficiently under limited networking, computing, and memory resources?* and *(2) How can partial PDSP be adapted for heterogeneous environments to effectively balance between policy specialization and generalization?*

Key Contributions. This paper addresses the questions above by making the following novel contributions:

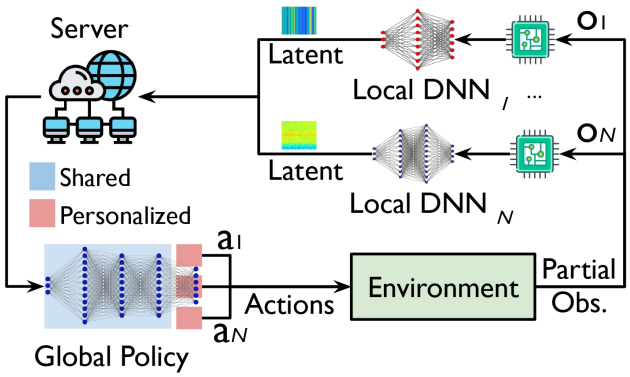

Figure 1: Overview of SDE−HARL.

• We introduce **S**calable **D**istributed Policy **E**xecution in **HARL** settings (in short, **SDE−HARL**). As shown in Figure 1, the core idea is to partition the policy deep neural network (DNN) of each agent into a lightweight *local* DNN, deployed on each mobile device, and a *global* DNN, executed at the nearest server. Each local DNN transforms each observation into a *"compressed"* latent representation by learning a prior distribution over the intermediate representation and minimizing its entropy during training, enabling adaptive compression through entropy coding. These latent representations are sent to the edge server, thus decreasing computation at the mobile device and networking overhead while retaining agent-specific behaviors;

• To utilize monotonic improvement theory, we propose two new key innovations. **First**, we introduce a novel PDSP paradigm for HARL systems in which the majority of the *global* DNN layers, are shared among agents with similar roles, while only the final layers are individualized to accommodate diverse action spaces, as shown in Figure 1. **Second**, we devise a new two-phase training procedure: (1) each agent's policy is independently updated to optimize the shared *global* policy, enabling the extraction of agent-invariant latent features; (2) we perform sequential policy updates on agent-specific output heads while the shared *global* DNN is frozen;

• New mechanisms for *role-aware diversity (RAD)* and *identity-aware diversity (IAD)* are developed for the shared part of *global* policy to promote behavioral diversity. RAD enables parameter sharing among agents with similar roles by encoding agent-specific behavioral dynamics into a latent space and clustering them based on their observations and reward function. IAD encourages agents within the same role group to exhibit diverse behaviors by maximizing the mutual information between and agent's identity and its individual trajectory;

• We integrate SDE−HARL into existing HARL algorithms, including HAPPO (Kuba et al. 2021), HATRPO (Kuba et al. 2021), and HASAC (Liu et al. 2023a). We compare the resulting variants against their original counterparts across several challenging datasets, namely Google Research Football (GRF) (Kurach et al. 2019), StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), and SMACv2 (Ellis et al. 2023). In addition, we prototype SDE−HARL using Raspberry Pi boards for agents and an RTX 8000 GPU acting as an edge server and computed the end-to-end execution latency and networking overhead. Experimental results indicate that SDE−HARL achieves 15% more win rate than state-of-the-art (SOTA) baselines at 3×10^6 time steps. SDE−HARL also decreases latency, energy consumption, and networking overhead by $2\times$, $2.5\times$, and $5\times$, respectively compared to (Kuba et al. 2021; Liu et al. 2023a; Zhong et al. 2024).

Background

Cooperative MARL. A fully-cooperative multi-agent task can be formulated as a Dec-POMDP (Oliehoek and Amato 2016), which is defined as a tuple $\mathcal{G} = \langle N, S, A, P, R, O, \Omega, n, \gamma \rangle$, where N is a finite set of n agents, $s \in S$ is the true state of the environment, A is the set of actions, and $\gamma \in [0, 1)$ is a discount factor. At each time step, each agent $i \in N$ obtains its own observation $o_i \in \Omega$ according to the observation function $O(s, i)$, and selects an action $a_i \in A$, which results in a joint action vector $\mathbf{a}$ for all agents together. The environment then transitions to a new state s' based on the transition function $P(s' \mid s, \mathbf{a})$, and induces a global reward $r = R(s, \mathbf{a})$ shared by all agents. Each agent has its own action-observation history $\tau_i \in \mathcal{T}_i \doteq (\Omega_i \times A)^*$. Due to partial observability, each agent conditions its policy $\pi_i(a_i \mid \tau_i)$ on τ_i. The joint policy $\boldsymbol{\pi}$ induces the joint action-value function $Q^{\boldsymbol{\pi}}_{\text{tot}}(s, \mathbf{a}) = \mathbb{E}_{s_{0:\infty}, \mathbf{a}_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, \mathbf{a}_0 = \mathbf{a}, \boldsymbol{\pi} \right]$.

Rate-Performance in Lossy Compression. General data compression maps each possible data point to a variable-length symbol sequence for storage or transmission, and inverts the mapping on the receiver side. The optimal number of bits needed to store a discrete-valued random variable Z is given by the Shannon entropy. Entropy coding techniques are then used for compression (Rissanen and Langdon 1981; Berman, Karpinski, and Nekrich 2002). The entropy is also referred to as the bit rate R of the compression method. Task-oriented compression introduces an inherent trade-off between R and the task-specific performance P based on the latent representation z for a downstream task:

$$\hat{z}^* = \arg\min_{\hat{z}} \sum_{x \in \mathcal{D}} P(\hat{z}) + \lambda R(\hat{z}), \tag{1}$$

where $\hat{z}$ is lossy discrete representation of input x, and λ controls the trade-off between performance and bit-rate.

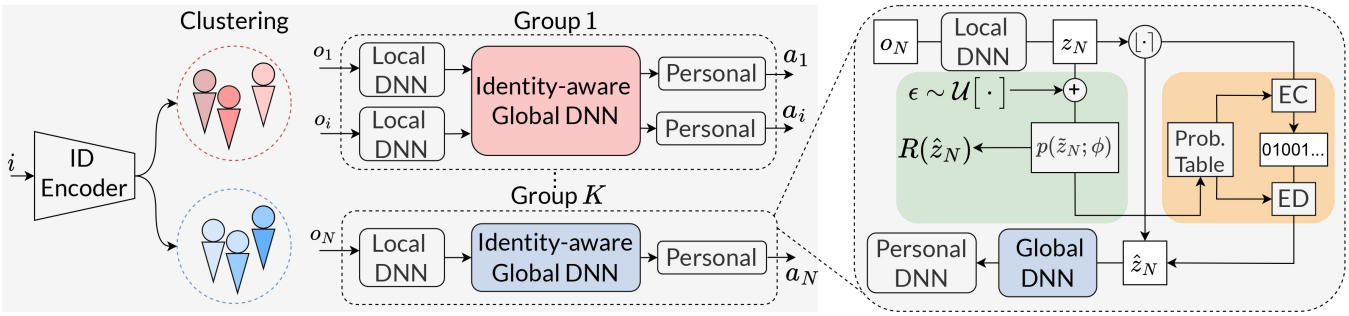

Figure 2: Overview of the proposed SDE−HARL framework. Agents are grouped by learned identity representations and decomposed into local, global, and personal DNN modules. Shared global DNNs are trained within each group. The right shows the training (green) and execution (orange) phases of the compression-aware policy pipeline.

SDE−HARL Architecture

Figure 2 illustrates the architecture of our proposed framework. Given a system of N agents with observations $O_{\mathcal{N}} = (o_1, o_2, \ldots, o_N)$, each agent i processes its observation o_i by an agent-specific local DNN $f_i(\cdot)$ to obtain a latent representation z_i. These local DNNs are independently parameterized to capture heterogeneous agent-specific properties ($e.g.$, different sensor modalities or observation dimensions). The resulting features are then quantized into $\hat{z}_i$ and transmitted to a nearby edge server. On the server side, all encoded representations $\hat{z}_i$ are clustered into K groups via RAD mechanism (see Section). Each group $k \in 1, \ldots, K$ contains M_k agents that share an identity-aware global DNN $f_k^S(\cdot)$, enabling the extraction of features z_m^k with $m \in M_k$. These features are then refined through personalized DNNs $f_{k,m}^P(\cdot)$, which produce the final actions a_i, forming the overall policy π_ξ^i for each agent. The computation procedure is:

$$
\begin{aligned}
z_i &= f_i(o_i; \theta_i), && \forall i \in \mathcal{N}, \\
\hat{z}_i &= Q(z_i), && \forall i \in \mathcal{N}, \\
z_m^k &= f_k^S(\hat{z}_i; \phi_k), && \forall k \in \mathcal{K}, \forall m \in \mathcal{M}_k, \\
a_i &\sim \pi_{\xi_i}^i(a_i \mid o_i), && \text{where } \pi_{\xi_i}^i(a_i \mid o_i) = \mathcal{G}(f_{k,m}^P(z_m^k; \psi_m^k)).
\end{aligned}
$$

In the equation above, θ_i, ϕ_k, and ψ_m^k are respectively the learnable parameters of the local DNN, the shared global DNN for group k, and the personalized DNN for agent m within group k. Moreover, ξ_i is the complete set of policy-making parameters $(\theta_i, \phi_k, \psi_m^k)$, while $\mathcal{G}(\cdot)$ is a method-specific mapping from network output to action.

Latent Feature Compression

We leverage Equation (1) to compress the latent features $\hat{z}_i$. In our case, the performance term translates to the cooperative reward obtained from the environment after taking a joint action by all agents. A key issue is that the quantization operation, which maps continuous embeddings z_i to discrete codes $\hat{z}_i$, is non-differentiable, and thus does not allow the optimization of parameters through backpropagation. To address this, we adopt a training strategy based on neural image compression (Ballé, Laparra, and Simoncelli 2016; Ballé et al. 2018). Specifically, during training – see right side of Figure 2 – we approximate quantization by injecting additive uniform noise, i.e., $\tilde{z}_i = z_i + e$, where $e \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$. This relaxation enables a differentiable approximation to the rate term, modeled as the negative log-likelihood of $\tilde{z}_i$ under a learnable prior $p_{\phi_i}(\tilde{z}_i)$, typically implemented as a factorized density model as $R(\hat{z}_i) \approx \mathbb{E}_{\tilde{z}_i}(-\log p_{\phi_i}(\tilde{z}_i))$.

For the performance term $P(\hat{z})$, we apply rounding $\hat{z}_i = \lfloor z_i \rceil$ to enforce discretization. Since rounding also yields a zero gradient, we use the straight-through estimator, replacing its gradient with the identity function to allow the gradient to flow backward during optimization. This allows the local DNN to learn discrete, compressible representations while maintaining task-relevant information. Therefore, the optimization objective in Equation (1) becomes:

$$
\theta_i^*, \phi_i^* = \arg\min_{\theta_i, \phi_i} \mathbb{E}_{o_i \sim \Omega,\, e \sim \mathcal{U}} \Big[\underbrace{-P\big(\lfloor f_i(o_i; \theta_i) \rceil\big)}_{\text{Policy Performance}} + \lambda \cdot \underbrace{\Big(-\log p_{\phi_i}\big(f_i(o_i; \theta_i) + e\big)\Big)}_{\text{Bit-Rate}} \Big]. \tag{2}
$$

Selective Parameter Sharing

Role-Aware Diversity. To encourage agent-specific behavior, we learn a latent identity representation z_i^{ID} for each agent i (Christianos et al. 2021). Specifically, we define an encoder $f_E^{\text{ID}}(\cdot)$ that maps each agent to a posterior distribution $q(z_i^{\text{ID}} \mid i)$, and a decoder $f_D^{\text{ID}}(\cdot)$ that reconstructs future observations and rewards conditioned on the current state and latent identity z_i^{ID}, as presented in Appendix A.1. Thanks to the bottleneck, identity embeddings z_i^{ID} captures agent-specific information.

To achieve this, we assume that an agent's identity i can capture its specific transition dynamics and reward function. Furthermore, both the identity i and the transition tuple $\tau_t = (o_{t+1}, r_t, o_t, a_t)$ can be mapped into a shared latent space $\mathbf{Z}^{\text{ID}}$, via the respective posterior distributions $q(z_i^{\text{ID}}|i)$ and $p(z_i^{\text{ID}}|\tau_t)$. The primary objective is to learn the distribution $q(z_i^{\text{ID}}|i)$, which we model as a Gaussian with learnable parameters e, denoted by $q_e(z_i^{\text{ID}}|i) = \mathcal{N}(\mu_e, \Sigma_e; i)$. To achieve this, we adopt the variational inference (Ganguly and Earp 2021) to optimize the KL diver-

gence $D_{\mathrm{KL}}(q_e(z_i^{\mathtt{ID}}|i)\|p(z_i^{\mathtt{ID}}|\tau_t))$. This leads to the following evidence lower bound (ELBO) on the log-likelihood of the transition:

$$\log p(\tau_t) \geq \mathbb{E}_{z_i^{\mathtt{ID}} \sim q_e(z_i^{\mathtt{ID}}|i)} \left[\log p_u(\tau_t \mid z_i^{\mathtt{ID}})\right] - D_{\mathrm{KL}}\left(q_e(z_i^{\mathtt{ID}} \mid i) \,\|\, p(z_i^{\mathtt{ID}})\right), \quad (3)$$

where q_e and p_u correspond to the encoder $f_E^{\mathtt{ID}}(\cdot)$ and decoder $f_D^{\mathtt{ID}}(\cdot)$, respectively. The reconstruction term in (3) can be decomposed as:

$$\log p_u(\tau_t \mid z_i^{\mathtt{ID}}) = \log p_u(r_t, o_{t+1} \mid s_t, z_i^{\mathtt{ID}}) \log p(s_t \mid z_i^{\mathtt{ID}}) = \log p_u(r_t \mid o_{t+1}, s_t, z_i^{\mathtt{ID}}) + \log p_u(o_{t+1} \mid s_t, z_i^{\mathtt{ID}}) + c.$$

Since $s_t = (o_t, a_t)$ is independent of $z_i^{\mathtt{ID}}$, the last term is discarded. After pretraining the encoder, we apply k-means to the identity embeddings $z_i^{\mathtt{ID}}$ to group similar agents (see more details in the Appendix A.1). Clustering is performed once before policy training, with each group sharing a global policy network – see left and middle side of Figure 3.

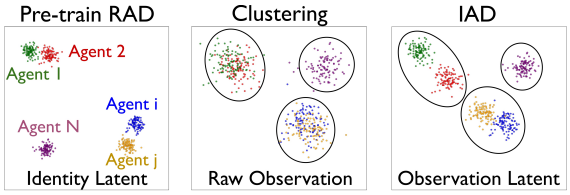

Figure 3: RAD and IAD mechanisms. Left: latent space of identity embeddings $z_i^{\mathtt{ID}}$ from the pre-trained identity encoder $f_E^{\mathtt{ID}}$. Middle: raw observation space, clustered based on RAD pre-training. Right: representation space produced by the identity-aware global DNN, which enhances inter-agent differentiation within the same group.

Identity-Aware Diversity. To encourage diverse behaviors within the shared global DNN (i.e., same group) – see right side of Figure 3 – we maximize the mutual information between the individual trajectory and agent's identity:

$$\mathcal{I}_{\boldsymbol{\pi}}(\tau_T; \mathtt{ID}) = H(\tau_T) - H(\tau_T \mid \mathtt{ID}) = \mathbb{E}_{\mathtt{ID}, \tau_T \sim \boldsymbol{\pi}} \left[\log \frac{p(\tau_T \mid \mathtt{ID})}{p(\tau_T)}\right], \quad (4)$$

where τ_T and $\mathtt{ID}$ are the random variables for the agent's local trajectory and identity, respectively. To make this objective tractable, we expand the trajectory distribution as $p(\tau_T) = p(o_0) \prod_{t=0}^{T-1} p(a_t \mid \tau_t) \cdot p(o_{t+1} \mid \tau_t, a_t)$ and $p(\tau_T \mid \mathtt{ID}) = p(o_0 \mid \mathtt{ID}) \prod_{t=0}^{T-1} p(a_t \mid \tau_t, \mathtt{ID}) \cdot p(o_{t+1} \mid \tau_t, a_t, \mathtt{ID})$. By substituting into Equation (4), the mutual information decomposes as:

$$\mathcal{I}_{\boldsymbol{\pi}}(\tau_T; \mathtt{ID}) = \mathbb{E}_{\mathtt{ID}, \tau_T} \Bigg[\underbrace{\log \frac{p(o_0 \mid \mathtt{ID})}{p(o_0)}}_{\text{(1) Initial Observation}} + \sum_{t=0}^{T-1} \underbrace{\log \frac{p(a_t \mid \tau_t, \mathtt{ID})}{p(a_t \mid \tau_t)}}_{\text{(2) Action Choices}} + \sum_{t=0}^{T-1} \underbrace{\log \frac{p(o_{t+1} \mid \tau_t, a_t, \mathtt{ID})}{p(o_{t+1} \mid \tau_t, a_t)}}_{\text{(3) Observation Transitions}} \Bigg]. \quad (5)$$

We omit the first term during optimization, as it depends on the environment's initial state distribution. For Policy-based methods, the second term quantifies the information gain about agent's action selection when the identity is given, which measures *"action-aware diversity"* as $I(a; \mathtt{ID} \mid \tau)$. To address this problem, we approximate the true conditional distribution $p(a_t \mid \tau_t, \mathtt{ID})$ using a parameterized policy $\pi(a_t \mid \tau_t, \mathtt{ID}) = \mathrm{SoftMax}(f_\xi(\tau_t, \mathtt{ID}))$ with $f_\xi(\cdot)$ as the identity-conditioned policy DNN. We then express the marginal distribution over actions as $p(a_t \mid \tau_t) = \sum_{\mathtt{ID}} p(\mathtt{ID} \mid \tau_t) \pi(a_t \mid \tau_t, \mathtt{ID})$. Following prior work (Sharma et al. 2019; Li et al. 2021), we assume that the posterior over identities is approximately uniform, i.e., $p(\mathtt{ID} \mid \tau_t) \approx p(\mathtt{ID})$ (assumption' justification is provided in the Appendix A.2), which leads to $p(a_t \mid \tau_t) \approx \bar{\pi}(a_t \mid \tau_t) = \frac{1}{n} \sum_{\mathtt{ID}} \pi(a_t \mid \tau_t, \mathtt{ID})$, where n is the number of distinct agent identities. This approximation allows us to derive a tractable lower bound:

$$\mathbb{E}_{\mathtt{ID}, \tau} \left[\log \frac{p(a_t \mid \tau_t, \mathtt{ID})}{p(a_t \mid \tau_t)}\right] \geq \mathbb{E}_{\mathtt{ID}, \tau} \left[\log \frac{\pi(a_t \mid \tau_t, \mathtt{ID})}{\bar{\pi}(a_t \mid \tau_t)}\right], \quad (6)$$

where the inequality holds under the assumption that $p(\cdot \mid \tau_t, \mathtt{ID}) \approx \pi(\cdot \mid \tau_t, \mathtt{ID}))$, which enables a variational lower bound on mutual information. We maximize this lower bound to optimize Term (2). Inspired by variational inference (Ganguly and Earp 2021), we optimize a tractable lower bound for Term (3) by introducing a variational posterior $q_\varphi(o_{t+1} \mid \tau_t, a_t, \mathtt{ID})$:

$$\mathbb{E}_{\mathtt{ID}, \tau} \left[\log \frac{p(\cdot \mid \tau_t, a_t, \mathtt{ID})}{p(\cdot \mid \tau_t, a_t)}\right] \geq \mathbb{E}_{\mathtt{ID}, \tau} \left[\log \frac{q_\varphi(\cdot \mid \tau_t, a_t, \mathtt{ID})}{p(\cdot \mid \tau_t, a_t)}\right]. \quad (7)$$

Similar to the second term, the inequality holds because for any q_φ, the KL divergence $D_{\mathrm{KL}}\left(p(\cdot \mid \tau_t, a_t, \mathtt{ID}) \,\|\, q_\varphi(\cdot \mid \tau_t, a_t, \mathtt{ID})\right)$ is non-negative. Intuitively, optimizing Equation (7) elicits agents to have diverse observations and thus measures *"observation-aware diversity"* as $I(o_{t+1}; \mathtt{ID} \mid \tau, a)$. To tighten this lower bound, we minimize the KL divergence with respect to the parameters φ. The gradient for updating φ is:

$$\nabla_\varphi \mathcal{L}(\varphi) = \nabla_\varphi \mathbb{E}_{\tau, a, \mathtt{ID}} \left[D_{\mathrm{KL}}\left(p(\cdot \mid \tau, a, \mathtt{ID}) \| q_\varphi(\cdot \mid \tau, a, \mathtt{ID})\right)\right] = -\mathbb{E}_{\tau, a, \mathtt{ID}, o_{t+1}} \left[\nabla_\varphi \log q_\varphi(o_{t+1} \mid \tau, a, \mathtt{ID})\right]. \quad (8)$$

Based on the lower bounds shown in Equation (6) and Equation(7), we introduce intrinsic rewards to optimize the objective in Equation (4) to elicit agent-specific behavior:

$$r^I = \mathbb{E}_{\mathtt{ID}} \Bigg[\beta_2 \underbrace{D_{\mathrm{KL}}\left(\beta_1 \pi(\cdot \mid \tau_t, \mathtt{ID}) \,\|\, p(\cdot \mid \tau_t)\right)}_{\text{Action-aware Diversity}} + \beta_1 \underbrace{\log q_\varphi(o_{t+1} \mid \tau_t, a_t, \mathtt{ID}) - \log p(o_{t+1} \mid \tau_t, a_t)}_{\text{Observation-aware Diversity}} \Bigg], \quad (9)$$

where $\beta_1, \beta_2 \geq 0$ are scaling factors for the intrinsic rewards. Specifically, when $\beta_1 = 0$, we only optimize the entropy term $H(\tau_T)$ in the mutual information objective (in Equation (4)), while β_2 is used to adjust the importance of policy diversity compared with transition diversity.

For Value-based methods, following the prior work (Christianos et al. 2021), we optimize Equation (4) by replacing the ϵ-greedy policy with a Boltzmann softmax over Q-values, and define the intrinsic reward accordingly:

$$r^I = \mathbb{E}_{\texttt{ID}}\Big[\beta_2 D_{\text{KL}}\left(\text{SoftMax}(\beta_1 Q(\cdot \mid \tau_t, \texttt{ID})) \,\|\, p(\cdot \mid \tau_t)\right) + \beta_1 \log q_\varphi(o_{t+1} \mid \tau_t, a_t, \texttt{ID}) - \log p(o_{t+1} \mid \tau_t, a_t)\Big].$$

Training and Policy Update

Learning Objective. To jointly encourage well-coordination and behavioral diversity under parameter sharing, we augment the environment reward r^e with an identity-aware intrinsic reward r^I, scaled by a hyperparameter β. Thus, the learning objective in Equation (2) is refined as follows:

$$\mathcal{L}(\xi_i, \phi_i) = \underbrace{-\mathbb{E}_{\boldsymbol{\pi}}\left[r^e + \beta r^I\right]}_{\text{Combined Reward}} + \lambda \cdot \underbrace{R_{\phi_i}(\hat{z}_i)}_{\text{Bit-Rate}}, \tag{10}$$

and the policy gradient for updating ξ_i is:

$$\nabla_{\xi_i}\mathcal{L}(\xi_i) = -\mathbb{E}_{\boldsymbol{\pi}}\left[\nabla_{\xi_i}\log \pi_i(a_i \mid \tau_i; \xi_i) \cdot \left(\hat{A}_i + \beta \hat{r}^I\right)\right],$$

where $\boldsymbol{\pi}$ is the joint policy, and $\hat{A}_i$ denotes the advantage estimate for agent i.

Policy Update. Our training procedure consists of two distinct phases: (i) *shared representation learning phase*, where the agent policies $\pi^i_{\xi_i}$ are updated simultaneously, following the standard MARL paradigm. This allows training a shared global DNN $f^S_k(;\phi_k)$ that extracts agent-invariant latent features; and (ii) *policy fine-tuning phase*, in which the local DNN $f_i(;\theta_i)$ and shared global DNN $f^S_k(;\phi_k)$ are frozen and only the agent's personalized DNN $f^P_{k,m}(;\psi^k_m)$ is sequentially updated (Liu et al. 2023a) to ensure monotonic improvement under (Kuba et al. 2021) a trust region (TR) constraint. In particular, the policy of agent i at iteration k is updated as:

$$\pi^i_{k+1}(;\xi_i) = \arg\max_{\pi^i}\left[L^{1:i}_{\boldsymbol{\pi}_k}(\boldsymbol{\pi}^{1:m}_{k+1}, \pi^i) - C D^{\max}_{\text{KL}}(\pi^i_k, \pi^i)\right],$$

where C depends on the discount factor and maximal advantage, and $D^{\max}_{\text{KL}}$ is the TR bound. $\boldsymbol{\pi}^{1:m}_{k+1}$ denotes the previous agent policies are updated at iteration k. The policy parameters are $\xi_i = [\theta_i, \phi_k, \psi^k_m]$, where θ_i and ϕ_k are fixed and ψ^k_m is updated. The local surrogate advantage function is:

$$L^{1:i}_{\boldsymbol{\pi}_k}(\boldsymbol{\pi}^{1:m}_{k+1}, \pi^i) = \mathbb{E}_{s,\, \boldsymbol{a}_{1:m}\sim\boldsymbol{\pi}^{1:m}_{k+1},\, a^i\sim\pi^i}\left[A^i_{\boldsymbol{\pi}}(s, \boldsymbol{a}^{1:m}, a^i)\right],$$

agent n performs an update only if $L > 0$; otherwise, it retains its current policy if $L \leq 0$, thereby ensuring performance does not degrade.

Experimental Evaluation

In this section, to demonstrate the effectiveness of our proposed SDE−HARL, we integrate it into a range of HARL baselines, including HAPPO (Kuba et al. 2021), HATRPO (Kuba et al. 2021), and HASAC (Liu et al. 2023a). We compare the resulting variants against their original counterparts across several challenging datasets, namely GRF (Kurach et al. 2019), SMAC (Samvelyan et al. 2019), and SMACv2 (Ellis et al. 2023). We evaluate the inference times of SDE−HARL with the experimental setup shown in Figure 4, composed of three Raspberry Pi 5 units and an NVIDIA Quadro RTX 8000 GPU. We add additional experiments with a different experimental setup composed of more powerful mobile devices in Appendix A.3. Additional details about the experimental setup and hyperparameters are presented in Appendix A.3.

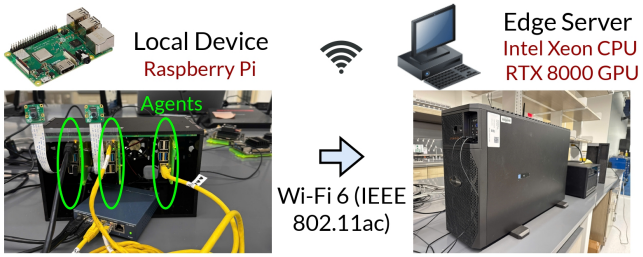

Figure 4: Experimental setup for evaluating latency and network overhead of SDE-HARL.

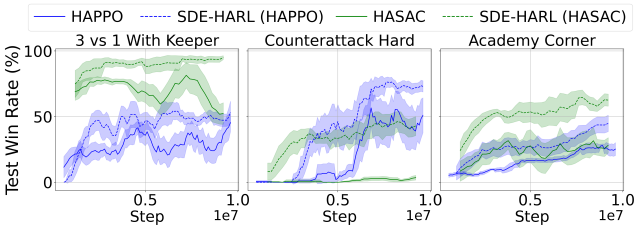

Figure 5: Comparison of SDE−HARL against baseline algorithms on the GRF dataset.

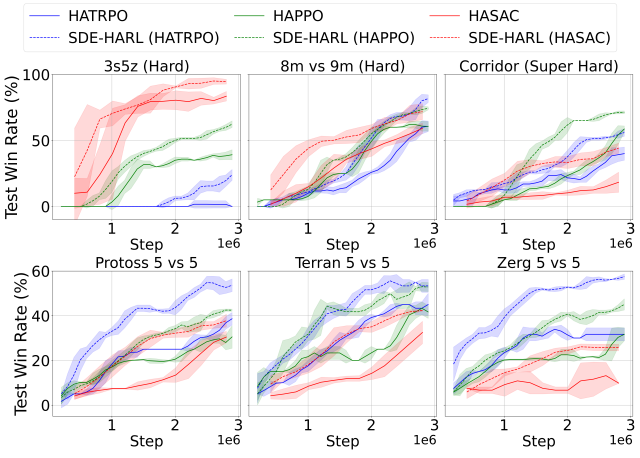

Figure 6: Comparison of SDE−HARL against baseline algorithms on SMAC (above) and SMACv2 (below) datasets.

Performance on GRF. We evaluate SDE−HARL on GRF dataset, as shown in Figure 5. These findings indicate that our method consistently outperforms the baselines. By leveraging partially PDSP, SDE−HARL is more efficient in discovering

Model	Reward ↑			
	GRF		SMAC/SMACv2	
	2 Agents	11 Agents	5 Agents	27 Agents
(i) No-Shared	1.18	1.09	11.71	11.85
(ii) Naive-Shared	1.24	1.06	11.82	10.65
(iii) No-IAD	1.38	1.32	13.62	13.41
(iv) No-Identity	1.32	1.27	14.26	13.15
(v) No-Action	1.42	1.44	15.67	15.48
(vi) No-Obs	1.36	1.35	14.75	15.23
(vii) No-RAD	<u>1.52</u>	1.55	<u>15.75</u>	15.64
(viii) CRL-Based	1.48	<u>1.57</u>	15.56	**16.77**
Ours	**1.58**	**1.62**	**16.89**	<u>16.72</u>

Table 1: Evaluation of SDE−HARL core components' contribution and comparison of its extension on GRF and SMAC/SMACv2 datasets: Best in **Bold** and second best in <u>underline</u>.

of coordinated behavior with respect to prior art. Notably, SDE−HARL (HASAC) achieves superior performance in challenging scenarios, such as Counterattack Hard.

Performance on SMAC and SMACv2. Figure 6 shows that SDE-HARL is consistently superior to the baselines in both sample efficiency and win rates. In challenging tasks such as SMAC 3s5z, SDE−HARL (HATRPO) achieves near-optimal performance, whereas its original version fails to learn the coordinator policy. Furthermore, our approach also achieves a substantial performance gap over the baselines in the hard Corridor task. Likewise, SDE−HARL (HATRPO) demonstrates significant gains across most SMACv2 environments. For example, our SDE−HARL (HATRPO) achieves a win rate of nearly 60% at 2.5×10^6 steps on the Protoss_5_vs_5 scenario, whereas its baseline only reaches around 30%.

Overall Insights. SDE-HARL achieves faster and more stable convergence than existing works by sharing representational knowledge across agents while preserving individual behavioral diversity. This advantage mitigates the impact of compressed observations, ensuring strong performance under limited communication and computation budgets.

Analytical Study

Ablation and Extension Studies. We conduct ablation studies to assess the contributions of the core components in the SDE−HARL framework. To this end, we deploy a series of modified architectures, each disabling a specific component as: (i) **No-Shared:** Standard HARL architecture with specific-agent policy DNN, (ii) **Naive-Shared:** Use PDSP without RAD and IAD; (iii) **No-IAD:** Use only RAD mechanism while ignoring IAD; (iv) **No-Identity:** Use RAD and only optimize the first term $H(\tau_T)$ in Equation (4) by setting β_1 in Equation (9) to zero; (v) **No-Action:** Use RAD and disable the *"action choices"* term in Equation (5) by setting β_2 in Equation (9) to zero; (vi) **No-Obs:** Use RAD and ablate the third item in Equation (5) by removing *"Observation-aware Diversity"* term in Equation (9); and (vii) **No-RAD**: Use fully IAD mechanism while disabling RAD. To validate our PDSP framework, we construct a variant that replaces our RAD and IAD mechanisms with a contrastive representation learning (CRL), called (viii) **CRL-based**. This variant builds on prior work (Liu et al. 2023b; Hu et al. 2023; Li et al. 2024), which represents the SOTA in PDSP for MARL. In contrast, recent approaches such as DiCo (Bettini, Kortvelesy, and Prorok 2024) and ADMN (Yu et al. 2024) are not directly compatible with our architecture, making their integration nontrivial and leading to sub-optimal performance. As a result, a direct comparison with these methods leads to unfairness. These experiments are conducted in both small and large-scale (i.e., number of agents) systems to evaluate the scalability and effectiveness of parameter sharing.

We first conduct ablation studies on small-scale scenarios (i.e., 2 and 5 agents) to identify which components of our method contribute most to the performance gains, as shown in Table 1. These findings indicate that **No-shared** performs the worst, even below **Naive-shared**, due to PDSP remaining effective in small-scale settings. Meanwhile, removing any part of our intrinsic reward leads to a noticeable decline in performance. Among these, the ablation of *"Action-aware Diversity"* in Equation 9 has the least impact. Notably, **CRL-based** also performs poorly in low-agent settings, achieving only a reward of 12.05 on GRF Run Pass and Shoot. This is because CRL relies on a sufficiently large set of negative samples to learn discriminative representations, which is difficult to achieve when the number of agents is limited. In contrast, **CRL-based** achieves competitive performance in large-agent settings (i.e., 10 and 27 agents), comparable to our SDE−HARL approach and surpassing its ablated variants.

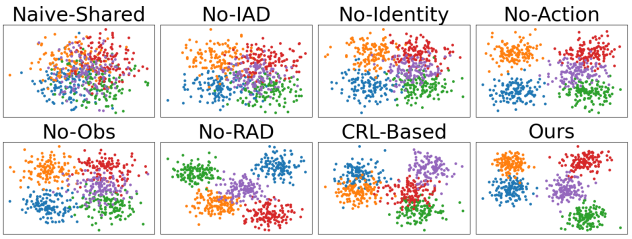

Figure 7: T-SNE plots of trajectory representations of different agents learned by different variants of SDE−HARL.

Understanding the Performance Disparities Across Methods. To do that, we visualize the learned trajectory representations using t-SNE on the SMAC 2s3z. Figure 7 indicates that our SDE−HARL produces well-separated clusters in the representation space, suggesting its capacity to learn disentangled agent-specific features. Notably, the Zealot agents (i.e., red, purple, and green) form a tight cluster, as do the Stalker agents (i.e., orange and blue), reflecting consistent role-based behavior patterns. These results highlight the effectiveness of our IAD and RAD mechanisms in enhancing representation diversity and cooperative performance.

Evaluating Performance-Resource Trade-off

We evaluate the trade-off between agent-side resource usage and task performance by varying the number of local DNN blocks n and adjusting the hyperparameter λ. As illustrated in the setup shown in Figure 4, each agent's policy network comprises 10 DNN blocks, with the computational split occurring at the n-th block. Figure 8 shows that deeper local DNN blocks enhance both compressibility (i.e., less

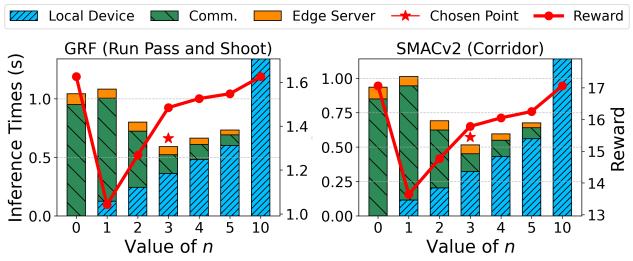

Figure 8: Evaluation of the inference time, and performance when changing the number of local DNN blocks n.

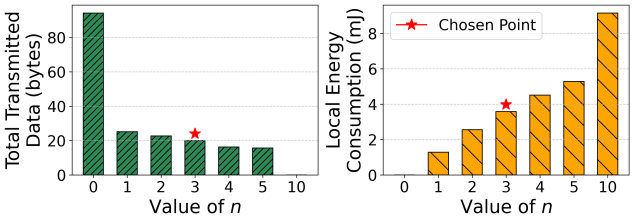

Figure 9: Local energy consumption and total transmitted data when changing the number of local DNN blocks n on GRF `Run_Pass_and_Shoot` scenario.

bandwidth demand) and performance through abstract feature extraction, albeit at the cost of increased computation on the local device. In contrast, shallow local models degrade both compressibility and policy quality.

To balance this trade-off, we select $n = 3$ for the remaining `SDE-HARL` settings, which significantly reduces inference time compared to traditional approaches, yielding $1.5\times$ reduction relative to fully server computation ($i.e., n = 0$) and a $2\times$ reduction compared to fully local computation (i.e., $n = 10$) with negligible performance degradation. Furthermore, setting $n = 3$ achieves a favorable trade-off, reducing total transmitted data by nearly $5\times$ and lowering the mobile device's energy consumption by approximately $2.5\times$ compared to the traditional configurations (i.e., $n = 0$ and $n = 10$), as shown in Figure 9. Notably, `SDE-HARL` without data compression (i.e., fully local computation, $n = 10$, or server-side computation, $n = 0$) yields optimal performance, but this comes at the cost of the highest resource demand. Figure 10 shows that setting λ to 10^{-5} strikes a good balance between bit-rate and reward, achieving 1.25 bits per dimension (bits/dim) and a reward of 16 on SMACv2 `Corridor`. Meanwhile, the setting $\lambda = 10^{-2}$ achieves impressive compressibility, but results in poor performance on both datasets.

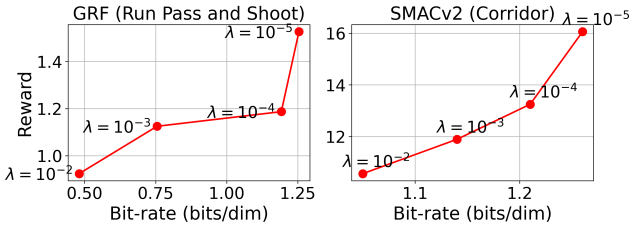

Figure 10: Evaluation of the trade-off between bit-rate and performance when changing the value λ in Equation (10).

Related Work

Distributed Neural Networks. Recent research has investigated partitioning strategies for DNNs (Zhang et al. 2025a, 2024, 2025b; Abdi et al. 2025). Since the first layers of most DNN tend to expand representations to facilitate feature extraction (Eshratifar, Abrishami, and Pedram 2021), compression is often required at the partitioning point (Matsubara et al. 2022b). To address this issue, several works introduce bottleneck injection techniques (Eshratifar, Esmaili, and Pedram 2019; Shao and Zhang 2020; Jankowski, Gündüz, and Mikolajczyk 2020), where feature compression is directly integrated into the backbone network without requiring separate compression modules. In (Matsubara et al. 2019), the bottleneck-injected model is fine-tuned using standard task losses such as cross-entropy for image classification. To mitigate performance degradation, later studies (Matsubara et al. 2020) incorporate knowledge distillation to recover the accuracy. For example, (Matsubara et al. 2022a) learn a prior distribution over intermediate representations and minimize its entropy during training, enabling adaptive compression through entropy coding. Building on this, `SDE-HARL` learns a prior over latent representations and applies an entropy-based coding to reduce the communication overhead.

Parameter Sharing in MARL. While PDSP may enhance scalability and efficiency, it hinders agent diversity and limits the emergence of sophisticated coordinated behaviors (Xu et al. 2023; Xu, Zhang, and Huang 2023). CDS (Li et al. 2021) adds agent-specific heads on top of a shared network, and SePS (Christianos et al. 2021) groups agents with shared parameters. However, these approaches scale poorly as the number of agents increases. Subsequently, ADMN (Yu et al. 2024) introduced a modular architecture with agent-specific routing mechanisms, while recent work (Liu et al. 2023b; Hu et al. 2023; Li et al. 2024) explored contrastive objectives to promote behavioral diversity. Recently, DiCo (Bettini, Kortvelesy, and Prorok 2024) proposed a policy decomposition framework that explicitly controls the level of diversity across different agents. However, these approaches are primarily designed for value-decomposition frameworks in homogeneous-agent systems, while current HARL methods often rely on an actor-critic approach. Additionally, existing PDSP approaches overlook the key characteristics of HARL algorithms (Zhong et al. 2024), namely *sequential update scheme* and *heterogeneous-agent TR*, and thus are not applicable to HARL settings. Our work is the first to introduce PDSP into HARL.

Conclusion

We introduced `SDE-HARL`, a novel HARL framework addressing the challenges of computational demand, bandwidth usage, and scalability. By decomposing the agent policy networks into lightweight local DNN and a shared global DNN, `SDE-HARL` significantly reduces the computational load on local devices while maintaining effective coordination among agents. Additionally, `SDE-HARL` leverages PDSP to enhance scalability and accelerate exploration during training through RAD and IAD mechanisms, which overcome key limitations of existing HARL methods. Experimental results demonstrate the effectiveness of our approach.

Acknowledgements

This work has been supported by the National Science Foundation under grants CNS-2312875 and OAC-2530896; by the Air Force Office of Scientific Research under grant FA9550-23-1-0261; by the Office of Naval Research under grant N00014-23-1-2221; and by the Defense Advanced Research Projects Agency under Cooperative Agreement D25AC00374-00.

References

Abdi, M.; Haque, K. F.; Meneghello, F.; Ashdown, J.; and Restuccia, F. 2025. PhyDNNs: Bringing Deep Neural Networks to the Physical Layer. In *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, 1–10.

Ballé, J.; Minnen, D. C.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. *ArXiv*, abs/1802.01436.

Ballé, J.; Laparra, V.; and Simoncelli, E. P. 2016. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, 1–5.

Berman, P.; Karpinski, M.; and Nekrich, Y. 2002. Approximating Huffman Codes in Parallel. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, 845–855. Berlin, Heidelberg: Springer-Verlag. ISBN 3540438645.

Bettini, M.; Kortvelesy, R.; and Prorok, A. 2024. Controlling behavioral diversity in multi-agent reinforcement learning. *arXiv preprint arXiv:2405.15054*.

Canese, L.; Cardarilli, G. C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; and Spanò, S. 2021. Multi-Agent Reinforcement Learning: A Review of Challenges and Applications. *Applied Sciences*, 11(11).

Christianos, F.; Papoudakis, G.; Rahman, M. A.; and Albrecht, S. V. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, 1989–1998. PMLR.

Du, W.; and Ding, S. 2021. A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. *Artif. Intell. Rev.*, 54(5): 3215–3238.

Ellis, B.; Cook, J.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J. N.; and Whiteson, S. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Eshratifar, A. E.; Abrishami, M. S.; and Pedram, M. 2021. JointDNN: An Efficient Training and Inference Engine for Intelligent Mobile Cloud Computing Services. *IEEE Transactions on Mobile Computing*, 20(2): 565–576.

Eshratifar, A. E.; Esmaili, A.; and Pedram, M. 2019. BottleNet: A Deep Learning Architecture for Intelligent Mobile Cloud Computing Services. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 1–6.

Ganguly, A.; and Earp, S. W. 2021. An introduction to variational inference. *arXiv preprint arXiv:2108.13083*.

Gronauer, S.; and Diepold, K. 2022. Multi-agent deep reinforcement learning: a survey. *Artif. Intell. Rev.*, 55(2): 895–943.

Hu, Z.; Zhang, Z.; Li, H.; Chen, C.; Ding, H.; and Wang, Z. 2023. Attention-guided contrastive role representations for multi-agent reinforcement learning. *arXiv preprint arXiv:2312.04819*.

Iqbal, S.; De Witt, C. A. S.; Peng, B.; Böhmer, W.; Whiteson, S.; and Sha, F. 2021. Randomized entity-wise factorization for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4596–4606. PMLR.

Jankowski, M.; Gündüz, D.; and Mikolajczyk, K. 2020. Joint Device-Edge Inference over Wireless Links with Pruning. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5.

Kuba, J. G.; Chen, R.; Wen, M.; Wen, Y.; Sun, F.; Wang, J.; and Yang, Y. 2021. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. *ArXiv*, abs/2109.11251.

Kurach, K.; Raichuk, A.; Stańczyk, P.; Zajac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; and Gelly, S. 2019. Google Research Football: A Novel Reinforcement Learning Environment. *ArXiv*, abs/1907.11180.

Li, C.; Wang, T.; Wu, C.; Zhao, Q.; Yang, J.; and Zhang, C. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3991–4002.

Li, T.; Zhu, K.; Li, J.; and Zhang, Y. 2024. Learning Distinguishable Trajectory Representation with Contrastive Loss. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 64454–64478. Curran Associates, Inc.

Li, T.; Zhu, K.; Luong, N. C.; Niyato, D.; Wu, Q.; Zhang, Y.; and Chen, B. 2022. Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 24(2): 1240–1279.

Liu, I.-J.; Jain, U.; Yeh, R. A.; and Schwing, A. 2021. Cooperative Exploration for Multi-Agent Deep Reinforcement Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6826–6836. PMLR.

Liu, J.; Zhong, Y.; Hu, S.; Fu, H.; Fu, Q.; Chang, X.; and Yang, Y. 2023a. Maximum Entropy Heterogeneous-Agent Reinforcement Learning. In *International Conference on Learning Representations*.

Liu, S.; Zhou, Y.; Song, J.; Zheng, T.; Chen, K.; Zhu, T.; Feng, Z.; and Song, M. 2023b. Contrastive identity-aware learning for multi-agent value decomposition. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*,

AAAI'23/IAAI'23/EAAI'23. AAAI Press. ISBN 978-1-57735-880-0.

Matsubara, Y.; Baidya, S.; Callegaro, D.; Levorato, M.; and Singh, S. 2019. Distilled Split Deep Neural Networks for Edge-Assisted Real-Time Systems. In *Proceedings of the 2019 Workshop on Hot Topics in Video Analytics and Intelligent Edges*, HotEdgeVideo'19, 21–26. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369282.

Matsubara, Y.; Callegaro, D.; Baidya, S.; Levorato, M.; and Singh, S. 2020. Head Network Distillation: Splitting Distilled Deep Neural Networks for Resource-Constrained Edge Computing Systems. *IEEE Access*, 8: 212177–212193.

Matsubara, Y.; Yang, R.; Levorato, M.; and Mandt, S. 2022a. Supervised Compression for Resource-Constrained Edge Computing Systems . In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 923–933. Los Alamitos, CA, USA: IEEE Computer Society.

Matsubara, Y.; Yang, R.; Levorato, M.; and Mandt, S. 2022b. SC2 Benchmark: Supervised Compression for Split Computing. *Trans. Mach. Learn. Res.*, 2023.

Nguyen, T. T.; Nguyen, N. D.; and Nahavandi, S. 2020. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*, 50(9): 3826–3839.

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated, 1st edition. ISBN 3319289276.

Oroojlooy, A.; and Hajinezhad, D. 2022. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11): 13677–13722.

Rissanen, J.; and Langdon, G. 1981. Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1): 12–23.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.-M.; Torr, P. H. S.; Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043.

Shao, J.; and Zhang, J. 2020. BottleNet++: An End-to-End Approach for Feature Compression in Device-Edge Co-Inference Systems. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 1–6.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2019. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*.

Tahir, N.; and Parasuraman, R. 2025. Edge Computing and Its Application in Robotics: A Survey. *Journal of Sensor and Actuator Networks*, 14(4).

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020a. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*.

Wang, Y.; Han, B.; Wang, T.; Dong, H.; and Zhang, C. 2020b. Off-policy multi-agent decomposed policy gradients. *arXiv preprint arXiv:2007.12322*.

Xu, P.; Zhang, J.; and Huang, K. 2023. Exploration via joint policy diversity for sparse-reward multi-agent tasks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23. ISBN 978-1-956792-03-4.

Xu, Z.; Zhang, B.; Li, D.; Zhang, Z.; Zhou, G.; Chen, H.; and Fan, G. 2023. Consensus learning for cooperative multi-agent reinforcement learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press. ISBN 978-1-57735-880-0.

Ye, D.; Zhang, M.; and Yang, Y. 2015. A Multi-Agent Framework for Packet Routing in Wireless Sensor Networks. *Sensors*, 15(5): 10026–10047.

Yu, Y.; Yin, Q.; Zhang, J.; Xu, P.; and Huang, K. 2024. ADMN: agent-driven modular network for dynamic parameter sharing in cooperative multi-agent reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24. ISBN 978-1-956792-04-1.

Yuan, L.; Zhang, Z.; Li, L.; Guan, C.; and Yu, Y. 2023. A Survey of Progress on Cooperative Multi-agent Reinforcement Learning in Open Environment. *ArXiv*, abs/2312.01058.

Zhang, M.; Abdi, M.; Ashdown, J.; and Restuccia, F. 2025a. Adversarial attacks to latent representations of distributed neural networks in split computing. *Computer Networks*, 273: 111755.

Zhang, M.; Abdi, M.; Dasari, V. R.; and Restuccia, F. 2024. Semantic edge computing and semantic communications in 6g networks: A unifying survey and research challenges. *arXiv preprint arXiv:2411.18199*.

Zhang, M.; Abdi, M.; Rifat, S.; and Restuccia, F. 2025b. Resilience of Entropy Model in Distributed Neural Networks. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 423–440. Cham: Springer Nature Switzerland. ISBN 978-3-031-72664-4.

Zhong, Y.; Kuba, J. G.; Feng, X.; Hu, S.; Ji, J.; and Yang, Y. 2024. Heterogeneous-agent reinforcement learning. *J. Mach. Learn. Res.*, 25(1).