KD-MARL: Resource-Aware Knowledge Distillation in Multi-Agent Reinforcement Learning

Monirul Islam Pavel*, Siyi Hu†, Muhammad Anwar Ma'sum*,
Mahardhika Pratama*(*Senior Member, IEEE*), Ryszard Kowalczyk*‡, Zehong Jimmy Cao*
*School of Computer Science and Information Technology, Adelaide University, Australia
†School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Australia
‡Systems Research Institute, Polish Academy of Sciences, Poland
Corresponding author: monirulislam.pavel@adelaide.edu.au

***Abstract*—Real-world deployment of multi-agent reinforcement learning (MARL) systems is fundamentally constrained by limited compute, memory, and inference time. While expert policies achieve high performance, they rely on costly decision cycles and large-scale models that are impractical for edge devices or embedded platforms. Knowledge distillation (KD) offers a promising path toward resource-aware execution, but existing KD methods in MARL focus narrowly on action imitation, often neglecting coordination structure and assuming uniform agent capabilities. We propose resource-aware Knowledge Distillation for Multi-Agent Reinforcement Learning (KD-MARL), a two-stage framework that transfers coordinated behavior from a centralized expert to lightweight, decentralized student agents. The student policies are trained without critic, relying instead on distilled advantage signals and structured policy supervision to preserve coordination under heterogeneous and limited observations. Our approach transfers both action-level behavior and structural coordination patterns from expert policies with supporting heterogeneous student architectures, allowing each agent's model capacity to match its observation complexity, which is crucial for efficient execution under partial or limited observability along with limited onboard resources. Extensive experiments on SMAC and MPE benchmarks demonstrate that KD-MARL achieves high performance retention while substantially reducing computational cost, retaining over 90% of expert performance while reducing computational cost by up to $28.6\times$ FLOPs. The proposed approach preserves expert-level coordination through structured distillation, enabling practical MARL deployment across resource-constrained onboard platforms.**

***Index Terms*—Knowledge Distillation, MARL, Resource-Aware Learning, Edge Intelligence, Compression, Knowledge Transfer**

I. Introduction

Multi-agent reinforcement learning (MARL) enables coordination among autonomous agents across domains including robotics [30], distributed sensing [12], and autonomous systems. Despite successes, deploying MARL in resource-constrained environments such as embedded systems, satellite networks remains challenging due to stringent requirements: low latency, limited memory, and real-time responsiveness.

Unlike computer vision or language modeling where compression targets network parameters, MARL's computational burden stems from the *decision-making cycle* rather than model size [16, 33]. Continuous policy updates, non-stationary observations, and decentralized coordination substantially increase training and inference costs [23]. High-capacity MARL models achieve excellent cooperation but are computationally expensive and memory-intensive [7], limiting deployment in on-board or distributed systems where efficiency is critical.

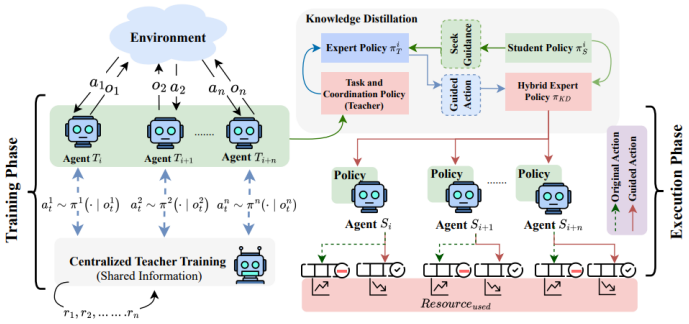

Fig. 1: Overview of proposed framework for teacher-student based CTDE inspired knowledge distillation in MARL, where the teacher learns expert coordination during training and guides lightweight student policy with near-expert performance & reduced resource cost.

Knowledge distillation addresses these constraints by transferring behavioral and structural knowledge from high-capacity *teacher* models to lightweight *student* agents [20, 24]. Unlike standard RL relying on sparse or delayed environmental rewards, KD leverages supervised expert trajectories, enabling faster convergence and improved stability. In MARL, this distills multiple knowledge forms: *action distributions* serving as soft targets to reduce inference complexity; *coordination dependencies* capturing inter-agent optimization; and *value structure* stabilizing distributed decision-making [32]. These representations allow students to emulate expert decisions whilst requiring fewer floating-point operations, enhancing efficiency and reducing latency.

Existing MARL distillation approaches remain limited. Many prior methods primarily emphasize per-agent policy imitation, with less explicit preservation of inter-agent relations and role specialization [10, 28], while others assume homogeneous agents with identical observations [1, 37], which is unrealistic for practical applications where agents differ in sensing capabilities and operate under partial or noisy observations [12, 30]. Moreover, most works neglect the compression-efficiency trade-off, often yielding reduced policy diversity and suboptimal coordination when scaled down [19, 31].

Reformulating cooperative MARL as single-agent RL by treating joint state-action spaces as one virtual decision-maker

leads to centralized training and centralized execution (CTCE) [9, 36]. However, this encounters scalability challenges as joint spaces grow exponentially with agent count. MARL also faces partial and heterogeneous observations: agents perceive environments differently based on roles or locations, requiring models accommodating varying input complexity [37]. Our framework introduces heterogeneous student architectures, assigning smaller models to agents with simpler inputs and larger models to those with richer observations, aligning model expressiveness with input complexity [14, 19] to avoid unnecessary computational overhead.

We propose KD-MARL, a two-stage knowledge distillation framework enabling multi-agent deployment under strict computational and memory constraints via centralized training, decentralized execution (CTDE) (Fig. 1). First, a high-capacity expert policy is trained using standard MARL algorithms; second, coordination knowledge is distilled into ultra-lightweight students for efficient operation on limited hardware [28]. Overall, the distillation objective integrates policy imitation, coordination structure preservation, and RL feedback [31], enabling students to retain expert cooperation while adapting to resource limitations and partial observability. The framework supports heterogeneous architectures, aligning each agent's capacity with observation complexity to minimize redundant computation [14, 37]. Our contributions are:

- Introduce KD-MARL, a two stage teacher-student framework that transfers coordinated behavior from a centralized MAPPO expert to lightweight decentralized agents.
- Design novel distillation loss combining action-policy fidelity, preservation for students pattern, and coordination under limited heterogeneous observations.
- Propose teacher-guided advantage distillation, where GAE-based advantage targets computed from the frozen expert critic replace environment-driven value learning, ensuring stable critic-free policy optimization.
- Achieve near-expert performance retention while reducing computational cost by up to $28.6\times$ in MPE and $11.7\times$ in SMAC in terms of FLOPs per episode, with corresponding improvements in inference-time throughput.

II. Related Works

In deep RL, knowledge distillation enables compressed models to maintain decision-making capabilities of larger counterparts. For multi-agent systems, this addresses inherent MARL scalability challenges. Early work focused on policy transfer under centralized training with decentralized execution. Czarnecki et al. [5] demonstrated distillation across heterogeneous action spaces, whilst Gao et al. [10] proposed KnowRU for structured knowledge reuse. However, these methods require high-quality teacher policies and assume full observability, limiting applicability in partially observable environments.

Recent work addresses these limitations through complementary approaches. Double Distillation Network [17] incorporates internal and external knowledge signals to improve coordination and exploration in sparse reward settings. For scenarios prohibiting online interaction, offline methods [29] enable distillation from static datasets, though balancing compression with policy expressiveness remains challenging.

Computational efficiency has motivated integrating distillation with network pruning. Liu et al. [21] introduced RL-guided compression dynamically removing redundant parameters under teacher supervision. Dan et al. [6] extended this with unified pruning and distillation. Domain-specific applications emerged, such as Chen et al.'s portfolio management system [3] implementing role-aware knowledge transfer for dynamic task allocation. Ensemble approaches [5] leverage multiple teachers but increase training complexity and may reduce policy diversity. Despite advances, current methods exhibit significant limitations. CTDS [36] distills Q-values without model compression or relational modeling between agents. CTPDE [25] achieves policy transfer but incurs high computational costs and produces homogeneous behaviors. Transformer-based approaches [28] offer feature-level distillation but require specialized architectures poorly generalizing to standard MARL algorithms. PTDE [4] introduces auxiliary modules for personalized representations, increasing model complexity. MAST [15] reduces computational load through pruning but lacks mechanisms ensuring behavioral consistency with teacher policies.

III. Methodology

We formulate cooperative MARL as a Dec-POMDP under CTDE, where training uses the joint state s and execution uses local observations o_i. In MARL, deploying trained agents in real-world environments presents challenges, particularly in resource-constrained settings due to complex models and large trial-and-error learning cycles, which are computationally expensive. These cycles involve continuous exploration and feedback through interaction with the environment, demanding significant computational resources that make deployment impractical in real-time applications with limited edge resources. To enable deployment in resource-constrained and real-time settings, we propose ***KD-MARL*** (Fig. 2), a two-stage training framework designed to reduce computational overhead and accelerate decision cycles. In the first stage, a high-capacity teacher policy π_T is trained under centralized training with decentralized execution, using full observations and a centralized critic $V_T(s)$ to capture joint-state value information. In the second stage, compact student policies π_S are trained without learning any critic. Instead, the critic's role is replaced through Teacher-Guided Advantage Distillation, where Distilled GAE Advantage Targets computed from the frozen teacher critic provide the policy-gradient signal for student optimization [34]. By eliminating critic inference and reducing network capacity at execution time, the resulting decentralized students achieve faster decision-making and lower computational and memory costs while retaining coordinated near-expert performance.

A. Model Architectures

Both teacher and student agents utilize recurrent neural network architectures with different hidden dimensions for

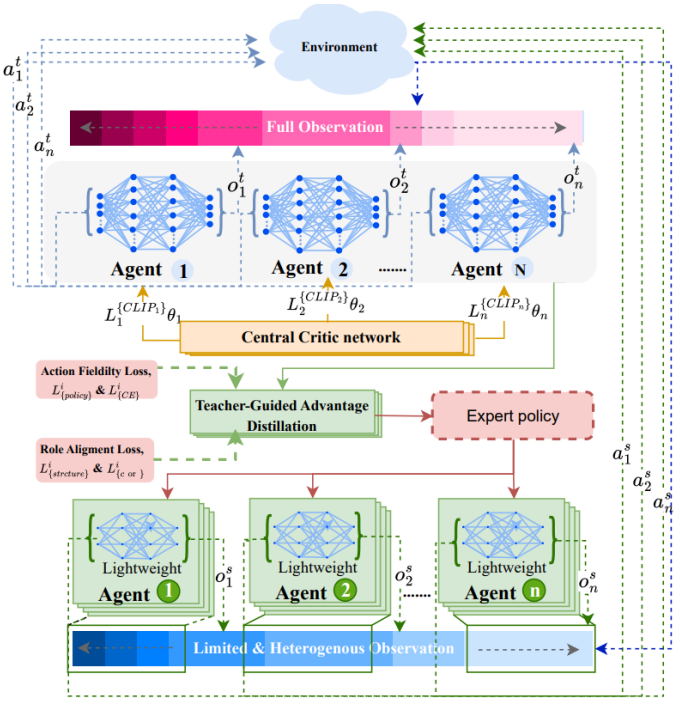

Fig. 2: Proposed KD-MARL architecture with two-stage training strategy in limited & heterogeneous setup.

partially observable multi-agent tasks. Teacher model are large, expressive networks trained offline with state-of-the-art MARL algorithms, capturing long-term dependencies and complex coordination. Student agents adopt structurally similar but smaller architectures with fewer recurrent units, enabling effective knowledge transfer whilst reducing complexity for resource-efficient onboard deployment.

Teacher Model. For each agent $i \in \{1, \ldots, N\}$, the teacher employs deep recurrent or feedforward networks with large hidden dimensions (256 or 512 units per layer). Input comprises comprehensive observations $\mathcal{O}_{T,i} \in \mathbb{R}^{d_{T,i}}$, where $d_{T,i}$ varies across agents due to heterogeneous roles and features. Output includes action probabilities $\pi_{T,i}(a|s) \in \Delta^{|\mathcal{A}_i|}$ for policy distillation and value estimates $V_{T,i}(s) \in \mathbb{R}$, where $\mathcal{A}_i$ is the agent-specific action space and $s \in \mathcal{S}$ the joint state.

Student Model. Students adopt compact architectures with reduced hidden dimensions (16 or 32) for deployment in resource-limited environments. Each agent i operates on constrained observations $\mathcal{O}_{S,i} \in \mathbb{R}^{d_{S,i}}$ where $d_{S,i} \ll d_{T,i}$. students access only positional limited features comparing to teacher agents. Observation spaces differ across student agents ($\mathcal{O}_{S,i} \neq \mathcal{O}_{S,j}$) to recreate heterogeneity tackle uncertainty. Despite reduced and agent-specific inputs, students produce task-consistent action distributions $\pi_{S,i}(a|s) \in \Delta^{|\mathcal{A}_i|}$, enabling faithful policy imitation under diverse constraints without requiring to learn from value function.

B. Two-stage Training Strategy with Teacher-Guided Advantage Distillation

The proposed KD-MARL framework adopts a two-stage training strategy (detailed in Algorithm 1) to decouple centralized learning from efficient decentralized execution. In the first stage, high-capacity teacher agents are trained using MAPPO under CTDE, as KD-MARL distills GAE-based advantage targets from a centralized critic $V_T(s)$, making an actor-critic teacher the most natural fit, using a single MAPPO teacher across tasks. Each teacher learns a decentralized policy π_T supported by this $V_T(s)$, where s denotes the joint environment state. The critic provides variance-reduced value estimates that enable stable learning of coordinated behaviour across agents. After convergence, the teacher policy and critic are frozen and used exclusively to generate supervision signals.

a) Critic-Free Student Training via Advantage Distillation.: In the second stage, student agents are trained without learning any critic or value function. Removing the critic directly in actor-critic methods typically leads to unstable policy updates due to high-variance gradients and poor temporal credit assignment. KD-MARL addresses this challenge through *Teacher-Guided Advantage Distillation*, where advantage targets distilled from the frozen teacher critic replace the role of the student critic. This preserves the stabilizing function of the critic while eliminating its computational and memory overhead during student training and execution.

b) Distilled GAE Advantage Targets.: When trajectories are sampled from the expert buffer, temporal-difference residuals are computed using the teacher critic δ_t^T through Eq. 1 where r_t is the shared team reward at time t and $\gamma \in (0, 1)$ is the discount factor. These residuals are accumulated using generalized advantage estimation to obtain low-variance, temporally consistent supervision from Eq. 2. The resulting distilled advantages A_T^{GAE} fully replace the student critic and serve as the sole advantage signal during policy optimization.

$$\delta_t^T = r_t + \gamma V_T(s_{t+1}) - V_T(s_t), \tag{1}$$

$$A_T^{\mathrm{GAE}}(t) = \sum_{l=0}^{L-1} (\gamma\lambda)^l \delta_{t+l}^T, \tag{2}$$

c) Advantage-Driven Policy optimization.: Student policies $\pi_S(a_t \mid o_{S,t})$, operating on restricted and heterogeneous local observations $o_{S,t}$, are optimized using a PPO-style objective shown in Eq.(3) where $r_t(\theta)$ indicates the probability ratio between the updated student policy and the behaviour policy that generated the expert data. Overall, ϵ constrains policy updates to ensure stable optimization.

$$r_t(\theta) = \frac{\pi_S(a_t \mid o_{S,t})}{\pi_{\mathrm{beh}}(a_t \mid o_{S,t})} \tag{3}$$

$$\mathcal{L}_{\mathrm{PPO}} = \mathbb{E}\left[\min\begin{pmatrix} r_t(\theta)\, A_T^{\mathrm{GAE}}(t), \\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, A_T^{\mathrm{GAE}}(t) \end{pmatrix}\right] \tag{4}$$

This advantage-guided optimization alone is insufficient to preserve coordinated multi-agent behavior under decentralized execution. To prevent behavioral drift and loss of coordination, policy optimization is regularized using explicit knowledge distillation. The distillation loss $\mathcal{L}_{\mathrm{KD}}$, aligns student and teacher action distributions, preserves inter-agent relational

structure, and maintains functional role specialization. The complete objective for Stage 2 student training is shown in Eq. (5) where $\mathcal{H}(\pi_S)$ is an entropy regularizer and ζ controls its contribution. No critic is learned or evaluated during this stage.

$$\mathcal{L}_{\text{Stage2}} = -\mathcal{L}_{\text{PPO}} + \mathcal{L}_{\text{KD}} - \zeta\,\mathcal{H}(\pi_S) \tag{5}$$

In Stage-2, student agents are trained without learning or maintaining any critic or value function. The centralized teacher critic is used only to compute GAE-based advantage targets offline, which replace the critic's role in policy optimization under a PPO objective with distillation regularization. As a result, the student relies solely on a decentralized actor, producing a lightweight policy suitable for resource-constrained onboard execution.

C. Resource-Aware Deployment.

We analyze the deployment efficiency of the distilled student policies under resource-constrained settings. evaluating inference-time FLOPs, model size, and execution cost to determine whether the proposed lightweight agents preserve strong task performance while reducing onboard computation. In order to enable resource-aware deployment, KD-MARL framework addresses decision-cycle complexity by providing soft targets from teacher distributions, enabling immediate action adjustment without trial-and-error [11] and compressing coordination knowledge efficiently [13, 18]. As a result, the student relies solely on a decentralized actor, producing a lightweight policy suitable for resource-constrained onboard execution. At deployment, all centralized components, including the teacher critic and expert buffer, are discarded. Execution relies exclusively on ultra-lightweight decentralized student policies operating on local observations. By replacing critic learning with teacher-guided advantage distillation and enforcing coordination preservation through $\mathcal{L}_{\text{KD}}$, KD-MARL achieves stable critic-free training, faster decision cycles, and substantially reduced computational and memory requirements, making it well suited for resource-aware MARL deployment in onboard and edge environments.

D. Distillation Loss

We propose a novel distillation loss function for resource-constrained MARL, enabling ultra-lightweight student agents to inherit both behavioral competence and coordination structure from high-capacity teachers. The proposed distillation loss $\mathcal{L}_{\text{KD}}$ integrates four complementary components, each targeting distinct aspects of expert knowledge transfer with hyperparameters α, β, $\lambda_{\text{structure}}$, and λ_{cor} controlling relative influence.

$$\mathcal{L}_{\text{KD}} = \sum_{i=1}^{N}\left[\alpha\,\mathcal{L}^i_{\text{policy}} + \mathcal{L}^i_{\text{CE}} + \lambda(\mathcal{L}^i_{\text{strcture}} + \mathcal{L}^i_{\text{cor}})\right] \tag{6}$$

1) Action-Policy Fidelity Based Loss: To encourage behavioral imitation, we minimize the Kullback–Leibler (KL) divergence between the expert teacher policy and the student policy, using the same policy arguments that appear in the total KD loss. This formulation corresponds to soft logit distillation,

Algorithm 1 KD-MARL Student Distillation with Teacher-Guided GAE (Critic-Free)

Require: Expert buffer $\mathcal{D}_{\text{exp}}$ with $(s, s', r, \{o^i_T, o^i_S, a^i, \text{logits}^i_T, \phi^i_T\}_{i=1}^N)$; frozen teacher π_T and critic V_T (full obs, h_T=256); students $\{\theta^i_S\}$ (limited hetero obs, $h_S \in \{16, 32\}$); aligners $\{g_i\}$; role projections U_T, U_S; $\gamma, \lambda, \tau, \epsilon$; weights $\alpha, \beta, \lambda_{\text{str}}, \lambda_{\text{role}}, \zeta$

1: **Stage 1 (Teacher, CTDE):** Train MAPPO teacher (π_T, V_T) on full observations; freeze and populate $\mathcal{D}_{\text{exp}}$.
2: **Stage 2 (Student, critic-free):**
3: **repeat**
4: Sample minibatch $\mathcal{B} \subset \mathcal{D}_{\text{exp}}$
5: Compute teacher-guided advantages A^{GAE}_T from V_T (no student critic)
6: **for** each agent i=1, . . . , N **do**
7: $\hat{o}^i \leftarrow g_i(o^i_S)$; $(\ell^i_S, \phi^i_S) \leftarrow \text{Student}(\hat{o}^i; \theta^i_S)$
8: $\pi^i_T \leftarrow \text{softmax}(\text{logits}^i_T/\tau)$; $\pi^i_S \leftarrow \text{softmax}(\ell^i_S/\tau)$
9: $\mathcal{L}^i_{\text{PPO}} \leftarrow \text{PPO-clip}(\pi^i_S, \pi^i_{\text{beh}}, A^{\text{GAE}}_T)$ (critic-free)
10: $\mathcal{L}^i_{\text{KL}} \leftarrow D_{\text{KL}}(\pi^i_T \,\|\, \pi^i_S)$; $\mathcal{L}^i_{\text{CE}} \leftarrow \text{CE}(\pi^i_T, \pi^i_S)$
11: $\rho^i_T, \rho^i_S \leftarrow \text{softmax}(U_T\phi^i_T/\tau), \text{softmax}(U_S\phi^i_S/\tau)$
12: **end for**
13: $\mathcal{L}_{\text{str}} \leftarrow \sum_{j<i}\big(\cos(\phi^i_T, \phi^j_T) - \cos(\phi^i_S, \phi^j_S)\big)^2$; $\mathcal{L}_{\text{role}} \leftarrow \sum_i D_{\text{KL}}(\rho^i_T \,\|\, \rho^i_S)$
14: $\mathcal{L} \leftarrow -\sum_i \mathcal{L}^i_{\text{PPO}} + \sum_i(\alpha\mathcal{L}^i_{\text{policy}} + \beta\mathcal{L}^i_{\text{CE}}) + \lambda_{\text{str}}\mathcal{L}_{\text{str}} + \lambda_{\text{role}}\mathcal{L}_{\text{role}} - \zeta\sum_i \mathcal{H}(\pi^i_S)$
15: Update $\{\theta^i_S\}$ and $\{g_i\}$ by gradient descent on $\mathcal{L}$ (teacher frozen)
16: **until** convergence
17: **return** decentralized students $\{\pi_{\theta^i_S}\}$ for onboard execution (critic discarded)

where the full action distribution (soft targets) from the teacher is transferred to the student. Accordingly, the KL-based action imitation loss for agent i is defined in Eq. (7). This ensures that the student's action probability distribution remains close to that of the teacher [38].

$$\mathcal{L}^i_{\text{policy}} = \mathbb{E}_{o^i_t \sim d^{\pi_S}}\left[D_{\text{KL}}\Big(\pi^i_T(\cdot \mid o_{T,i}) \,\|\, \pi^i_S(\cdot \mid o_{S,i})\Big)\right] \tag{7}$$

$$\begin{aligned}\mathcal{L}^i_{\text{CE}} &= \text{CE}\big(\pi^i_T(\cdot \mid o_{T,i}),\, \pi^i_S(\cdot \mid o_{S,i})\big) \\ &= \mathbb{E}_{o^i_t \sim d^{\pi_S}}\left[-\sum_a \pi^i_T(a \mid o_{T,i}) \log \pi^i_S(a \mid o_{S,i})\right]\end{aligned} \tag{8}$$

The second component is the Cross Entropy Loss (Eq. 8), which refines the distillation process by encouraging the student to select the same actions as the teacher. This corresponds to hard logit distillation, where the supervision emphasizes the teacher's most probable actions (hard targets). This loss function $\mathcal{L}^i_{\text{CE}}$ focuses on aligning student agent selects the same action as the teacher for the most probable choices, ensuring that the student not only imitates the distribution but also the teacher's specific choices [35].

2) Structure Relation and Coordinated Role based Loss: Beyond enabling per-agent action imitation, we define coordination as preserving inter-agent relational geometry and agent-

specific role specialization as effective multi-agent distillation requires retaining both dependencies between agents and their functional differentiation [29]. The expert policy produces latent embeddings ϕ_T^i that capture both individual behavioral features and inter-agent dependencies. The student embeddings ϕ_S^i must therefore retain these structural properties to support coordinated behavior under restricted observations.

To transfer the teacher's relational geometry, we minimize discrepancies in pairwise cosine similarity between teacher and student latent embeddings. For each agent pair (i, j), the structural relation loss $\mathcal{L}^i_{\text{structure}}$ is defined in Eq. (9) to ensure that student agents maintain consistent relational patterns even with reduced capacity and local observations.

$$\mathcal{L}^i_{\text{structure}} = \sum_{j<i} \Big[\cos\big(\phi_T^i,\, \phi_T^j\big) - \cos\big(\phi_S^i,\, \phi_S^j\big)\Big]^2 \tag{9}$$

To further maintain the coordinated behavior learned by the teacher, we include a coordinated role-based loss that aligns the role representations of the teacher and student. Moreover, it ensures that the student's latent role embedding ρ_S^i remains consistent with the teacher's role embedding ρ_T^i, preserving role-specific contributions that are essential for cooperative multi-agent decision-making. The coordinated role-based distillation loss is defined in Eq.(10) where the role distributions are computed via ρ_T^i and ρ_S^i.

$$\mathcal{L}^i_{\text{corr}} = D_{\text{KL}}\Big(\rho_T^i \parallel \rho_S^i\Big), \tag{10}$$

$$\rho_T^i = \text{softmax}\left(\frac{U_T \phi_T^i}{\tau}\right), \qquad \rho_S^i = \text{softmax}\left(\frac{U_S \phi_S^i}{\tau}\right) \tag{11}$$

Here, U_T and U_S denote the respective role projection matrices, and τ controlling distribution sharpness. Overall, this coordinated role-based loss $\mathcal{L}^i_{\text{role}}$ prevents the collapse of agent-specific roles during distillation, ensuring that the student agent continues to perform its designated role within the multi-agent system.

IV. Experiments

The experiments were conducted using the EPyMARL and PyMARL2 libraries, which serve as a comprehensive platform for integrating and managing various multi-agent environments. This library enables seamless execution of experiments across the SMAC & MPE environments, allowing for efficient simulation of agent interactions, reward structures, and learning processes.

A. Evaluation Environments

1) SMAC: We first evaluate KD-MARL on the StarCraft Multi-Agent Challenge (SMAC) using the standard maps 3m, 5m_vs_6m, 8m, and 3s5z [8]. A fully-observant teacher policy is trained to convergence and subsequently used to guide student policies operating under heterogeneous observation constraints. To emulate resource limitations in realistic multi-agent systems, each student agent receives only a subset of feature blocks that includes O (own), A (ally), E (enemy); while the remaining dimensions are masked to zero. For example, in 5m_vs_6m, Agents 0-1 observe enemy features only, Agent 2 observes enemy+own+ally, Agent 3 observes ally-only, and Agent 4 observes ally+own. Equivalent masking strategies are consistently applied across other SMAC maps (shown in Table I's Group and Feature blocks). This setup reflects practical constraints where agents cannot access full situational awareness due to sensing or processing bottlenecks.

2) MPE: We further test KD-MARL in the Multi-Agent Particle Environment (MPE) [22], where each agent normally observes an 18 dimensional feature vector. To simulate hardware-limited sensing, student agents are restricted to a randomly sampled subset of 8-10 features per episode, with the remaining inputs set to zero. The teacher is trained with full observations, and its policy is distilled to guide constrained students. This design enables the assessment of whether knowledge distillation can effectively transfer expert

TABLE I: Experimental Results: Algorithm Comparison with KD-MARL on SMAC with Limited & Heterogeneous observations. All algorithms use three configurations: FO (Full Observation: 109,880 params, hidden dim 256), LH (Limited Heterogeneous: 3,960 params, hidden dim 32), and LH+A (LH with heterogeneous architecture, hidden dim $\in [16, 32]$).

Map	Groups	Features	Metric	MAPPO			QMIX			VDN			KD-MARL	
				FO	LH	LH+A	FO	LH	LH+A	FO	LH	LH+A	LH	LH+A
3m	(0), (1), (2)	E, A, A+O	Return (/20)	$19.8_{\pm 0.2}$	$18.2_{\pm 0.4}$	$15.0_{\pm 0.7}$	$19.6_{\pm 0.3}$	$16.0_{\pm 0.6}$	$12.5_{\pm 0.8}$	$18.0_{\pm 0.5}$	$13.5_{\pm 0.6}$	$9.0_{\pm 0.7}$	**18.6**$_{\pm 0.4}$	**18.0**$_{\pm 0.5}$
			Win rate (%)	98.12	92.65	80.34	98.77	86.34	70.27	85.42	68.31	52.12	**94.78**	**90.39**
			TPS (ms)	6.5±0.3	6.2±0.4	6.3±0.4	6.6±0.3	5.9±0.4	**3.8±0.2**	6.0±0.3	**5.4±0.3**	4.3±0.2	5.5±0.3	4.1±0.2
8m	(0,1,2), (3,4), (5), (6), (7)	E, A, E+O, A+O, A+E	Return (/20)	$17.0_{\pm 1.3}$	$14.0_{\pm 0.7}$	$10.0_{\pm 1.2}$	$16.0_{\pm 0.5}$	$12.5_{\pm 0.9}$	$8.5_{\pm 1.1}$	$15.0_{\pm 0.8}$	$10.0_{\pm 0.9}$	$6.0_{\pm 1.0}$	**17.8**$_{\pm 0.6}$	**17.6**$_{\pm 0.4}$
			Win rate (%)	89.91	77.82	60.07	92.19	64.78	48.13	75.32	52.11	33.05	**88.97**	**88.23**
			TPS (ms)	21.5±1.2	22.0±1.3	21.8±1.2	21.9±1.0	18.1±1.1	15.8±0.9	19.0±1.0	**17.2±1.0**	**15.0±0.3**	17.3±0.9	**15.0±0.3**
5m_vs_6m	(0,1), (2), (3), (4)	E, E+O+A, A, A+O	Return (/20)	$18.0_{\pm 0.6}$	$16.5_{\pm 0.5}$	$13.0_{\pm 0.7}$	$19.1_{\pm 0.3}$	$14.0_{\pm 0.8}$	$10.0_{\pm 1.0}$	$16.0_{\pm 0.7}$	$11.0_{\pm 0.8}$	$7.0_{\pm 0.9}$	**16.8**$_{\pm 0.5}$	**16.5**$_{\pm 0.25}$
			Win rate (%)	61.85	58.09	44.78	58.93	50.12	38.79	50.10	38.22	25.14	**58.66**	**56.15**
			TPS (ms)	12.0±0.6	14.0±1.0	12.7±0.7	12.3±0.5	10.5±0.6	8.2±0.4	11.0±0.5	10.2±0.5	8.2±0.4	**10.0±0.5**	**8.0±0.3**
3s5z	(0,2), (1), (3,4),(5,6), (7)	E, E+A, A, A+O, E+O	Return (/20)	$18.5_{\pm 0.5}$	$16.8_{\pm 0.5}$	$13.5_{\pm 0.7}$	$18.7_{\pm 0.4}$	$15.0_{\pm 0.7}$	$10.5_{\pm 0.9}$	$16.5_{\pm 0.6}$	$11.0_{\pm 0.7}$	$7.0_{\pm 0.8}$	**17.2**$_{\pm 0.5}$	**16.5**$_{\pm 0.6}$
			Win rate (%)	68.31	55.66	42.54	60.48	50.12	36.95	53.42	40.33	24.12	**60.28**	**58.17**
			TPS (ms)	11.5±0.6	13.5±0.8	12.2±0.7	12.0±0.6	10.2±0.6	9.0±0.4	10.8±0.6	9.8±0.5	8.0±0.4	**9.7±0.5**	**7.9±0.4**

TABLE II: Experimental Results: Algorithm Comparison with KD-MARL on MPE with Limited & Heterogeneous observations. (SL = Speaker-Listerner, SS = Simple Spread, Adv = Adversary, L = Landmark, A = Agent, V = Velocity, P = Position, M = Message, D = Distance).

Map	Groups	Features	Metric	MAPPO			QMIX			VDN			KD-MARL	
				FO	LH	LH+A	FO	LH	LH+A	FO	LH	LH+A	LH	LH+A
SL	(0),	(L, V, P),	Return	-46.0±2.0	-82.0±3.2	-118.0±4.0	-55.0±3.0	-138.0±5.5	-205.0±7.2	-90.0±4.8	-170.0±6.5	-240.0±8.0	**-48.0±2.2**	**-50.0±2.4**
	(1)	(A(P), M, V, P)	TPS (ms)	6.0±0.3	5.3±0.4	4.0±0.3	4.8±0.3	4.6±0.3	4.2±0.3	4.7±0.3	4.5±0.3	4.3±0.3	**4.0±0.2**	**3.9±0.2**
SS	(0)	(L, A(P)),	Return	-46.0±2.1	-84.0±3.3	-120.0±4.1	-55.0±3.1	-142.0±5.8	-210.0±7.4	-92.0±4.9	-185.0±6.9	-248.0±8.6	**-48.5±2.1**	**-50.5±2.5**
	(1),(2)	V, P	TPS (ms)	8.1±0.5	7.3±0.6	4.5±0.4	4.9±0.3	4.6±0.3	4.3±0.3	5.1±0.3	5.0±0.3	4.7±0.3	**4.2±0.3**	**4.1±0.2**
Adv	(0)	(L, A(P)),	Return	-46.0±2.2	-86.0±3.4	-122.0±4.3	-55.0±3.2	-145.0±5.9	-215.0±7.6	-95.0±5.0	-190.0±7.0	-250.0±8.8	**-49.0±2.3**	**-51.0±2.6**
	(1), (2)	(A(P), V), D(A-A)	TPS (ms)	10.5±0.7	9.4±0.6	6.2±0.5	7.1±0.5	6.9±0.5	6.2±0.5	6.2±0.4	**6.0±0.3**	5.8±0.4	6.3±0.4	**4.9±0.3**

competence and maintain coordination performance under strict observation budgets.

B. Baselines and Comparisons

We evaluate KD-MARL against three established multi-agent reinforcement learning baselines: MAPPO [2], QMIX [26], and VDN [27]. Experiments are conducted in both the *StarCraft Multi-Agent Challenge (SMAC)* and the *Multi-Agent Particle Environments (MPE)*. Evaluations are carried out under three different settings that vary in the degree of agent heterogeneity and available observations:

- **FO (Full Observation):** All agents have access to the global state or equivalent complete information, representing the ideal coordination scenario.
- **LH (Limited Heterogeneity):** Agents receive only local, role-specific observations, introducing variation in perceptual access and coordination challenges.
- **LH+A (Limited Heterogeneity with Heterogeneous Architectures):** Similar to LH but with agents implemented using different network structures or capacities, simulating deployment on mixed hardware with varying resource constraints.

These baselines together cover a spectrum of centralized, partially factorized, and fully decomposed training schemes. The primary goal for comparison is to assess whether lightweight, decentralized agents trained via distillation can retain expert-level behavior with reduced computational and observational resources.

C. Results and Analysis

The experimental results for KD-MARL, compared against MAPPO baseline setup in both SMAC and MPE environments, highlight its ability to preserve expert-level performance while significantly reducing computational overhead.

a) ***Near-expert accuracy under heterogeneity.****:* KD-MARL performed close teacher under FO even when policies are compressed and observations are masked. On SMAC *3m*, the LH student attains around 94.0% of MAPPO teacher with FO and maintains a 94.78% win rate with only 3.34% drop. On the harder *8m* task, KD-MARL preserves win rate almost perfectly with 88.97% retention and 0.94% drop, while MAPPO trained directly under LH falls to 77.82% which is 12.09% less from its teacher policy. Similar retention is observed in coordination-heavy settings where on *5m_vs_6m*, KD-MARL gained 94.8% win-rate retention, whereas VDN under LH drops to 38.22%. Moreover, on *3s5z*, KD-MARL achieves 60.28%, outperforming QMIX-LH and VDN-LH. In MPE, under LH and LH+A, KD-MARL achieves returns within 4-6% of the FO MAPPO baseline across three maps, while QMIX and VDN suffer substantially larger degradations exceeding 20-40%. These gaps indicate that, under heterogeneous sensing, distillation primarily mitigates the coordination breakdown that limits non-KD baselines in constraints cases (LH, LH+A).

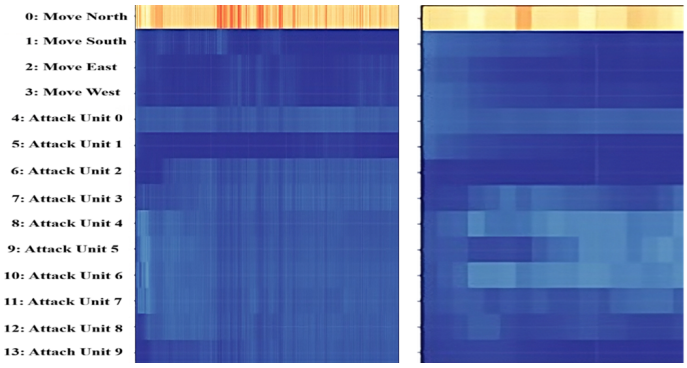

Fig. 3: The heatmaps show action-selection frequencies in the 3s5z SMAC scenario under constraints, with non-KD (left) and KD-MARL (right). Warmer colours indicate higher frequency. KD-MARL exhibits more concentrated and stable attack patterns, while the non-KD policy shows more dispersed actions, indicating reduced coordination.

b) ***Resource efficiency.****:* The retained performance is achieved with substantially lower compute demonstrating FLOPs (shown in Fig. 4) and time per step (TPS) reductions (shown in Tables I - II). In SMAC, FLOPs reductions range from $3.3\times$ to $11.7\times$ across maps. These savings translate into lower time consumption per step while maintaining near-teacher performance (within 4–6% in MPE), whereas QMIX and VDN deteriorate sharply under limited observations, highlighting KD-MARL's suitability for onboard deployment. In MPE, FLOPs are reduced by $26.7\times$ in *Speaker-Listener*, $30.0\times$ in *Simple Spread*, and $29.1\times$ in *Adversary*, yielding an average reduction of approximately $28.6\times$. Overall, lightweight computation leads to faster convergence per episode during student model execution with expert policy. In SMAC, runtime falls from 21.5 to 15.8 ms per episode on $8m$ with around 26% faster and 33% faster on 5m_vs_6m.

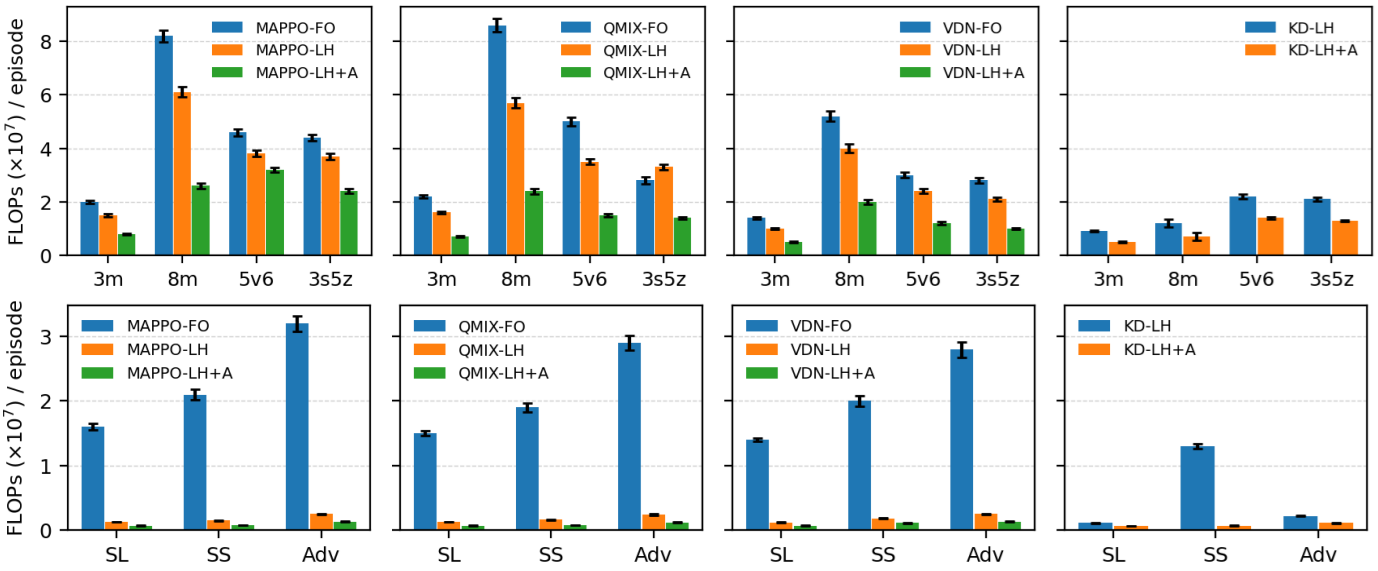

Fig. 4: FLOPs per episode comparison across SMAC and MPE maps, demonstrating resource-aware advantages of KD-MARL.

TABLE III: Comparison of different methods on 3s5z map (SMAC) in heterogeneous setup. (SL = Soft Logit, HL = Hard Logit, IR = Relation Loss, COR = Coordination Loss.)

Method	SL	HL	IR	COR	Agent Types	Win Rates (%)	FLOPs ($\times 10^7$)
[36]	•	-	-		RNN	46.08	1.13
[28]	•	•	-	-	Attention+RNN	54.92	3.05
[4]	•	-	•	-	Attention+RNN	61.56	3.17
Ours*	•	•	•	•	RNN	**58.17**	1.30 ± 0.03

Nevertheless, KD-MARL achieves this speedup with minimal performance cost, staying within 4-6% of the FO teacher on MPE benchmarks. By contrast, QMIX and VDN show steep degradation when observations are restricted, which makes KD-MARL a better fit for deployment scenarios where both speed and coordination quality are essential.

The heatmaps (in Fig. 3) illustrate the action selection frequency over time in the 3s5z StarCraft Multi-Agent Challenge scenario, comparing KD-MARL and non-KD based deployments. The outcomes demonstrate that the structure relation and role alignment losses preserve inter-agent coordination. In the KD-MARL setup, the action selection, particularly for attack commands (actions 4-13), shows more consistent and coordinated patterns, even under limited and heterogeneous observations. The warmer color intensity in KD-MARL indicates higher frequencies of coordinated actions, especially in attack, compared to the non-KD setup, where coordination is less consistent. This highlights how KD-MARL effectively preserves coordination patterns, ensuring better collective performance despite agent constraints.

Table III presents a broad comparison of performance retention across different approaches, showing win rates and FLOPs for various methods. Here, the performance retention is evaluated based on win percentages, where higher values indicate better retention of expert performance. Although other methods such as [36] and [28] show higher win rates, our approach demonstrates a better balance between performance retention and computational efficiency. Specifically, while methods like [4] achieve higher win rates, they also incur significantly higher computational costs in terms of FLOPs. In contrast, our method achieves a competitive win rate of around 58.17%, with $1.3 \pm 0.03 (\times 10^7)$ FLOPs, demonstrating superior performance retention with lower computational overhead. This confirms that our approach effectively balances both performance and resource efficiency, making it more suitable for real-world applications where computational resources are limited.

V. Conclusion

This work presented resource-aware KD-MARL, which is a comprehensive framework for achieving efficient and coordinated decision-making under strict computational and observational constraints. The study contributes through providing (i) two-stage distillation paradigm that transfers coordination knowledge from a high-capacity expert to ultra-lightweight, heterogeneous student agents,(ii) critic-free student optimization strategy based on distilled advantage signals and structured policy distillation, and (iii) extensive empirical validation across SMAC and MPE benchmarks. KD-MARL achieves near-expert performance retention of 92%, while reducing FLOPs by up to 96.5% and inference time by approximately 40%, confirming its capacity for real-time, decentralized execution in resource-limited systems. KD-MARL retains most of the teacher's performance, staying within 4–6% in MPE and demonstrating over 90% effectiveness across SMAC scenarios, while delivering major efficiency improvements. Specifically, we observe FLOPs reductions of up to $28.6\times$ in MPE and between $3.3\times$ and $11.7\times$ in SMAC, with throughput gains reaching 48% and 33% respectively. These findings indicate that KD-MARL successfully balances coordination quality with computational efficiency, making it practical for real-time deployment on resource-limited platforms. Future exten-

sions will focus on online adaptive distillation, multi-teacher transfer, and communication-efficient coordination, enabling broader applicability to autonomous and edge-level multi-agent systems.

ACKNOWLEDGMENTS

This work has been supported by the SmartSat CRC, whose activities are funded by the Australian Government's CRC Program. This work use an open-source realistic satellite simulator (Basilisk and BSK-RL) that is actively developed by Dr. Hanspeter Schaub and team at AVS Laboratory, University of Colorado Boulder. Also, the authors would like to express their sincere gratitude to BAE Systems for their invaluable support and collaboration throughout this research.

REFERENCES

[1] Bao, G., Ma, L., Yi, X., 2022. Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: A survey. Systems Science & Control Engineering 10, 539–551.

[2] Chen, L., Hu, B., Guan, Z.H., Zhao, L., Shen, X., 2021. Multiagent meta-reinforcement learning for adaptive multipath routing optimization. IEEE Transactions on Neural Networks and Learning Systems 33, 5374–5386.

[3] Chen, R., Lin, J., Zhou, Y., 2023. Portfolio management with multi-agent reinforcement learning: A role-aware distillation approach. Journal of Financial Data Science .

[4] Chen, Y., Mao, H., Mao, J., Wu, S., Zhang, T., Zhang, B., Yang, W., Chang, H., 2024. Ptde: personalized training with distilled execution for multi-agent reinforcement learning, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, pp. 31–39.

[5] Czarnecki, W.M., Pascanu, R., Osindero, S., Jayakumar, S., Swirszcz, G., Jaderberg, M., 2019. Distilling policy distillation, in: The 22nd international conference on artificial intelligence and statistics, PMLR. pp. 1331–1340.

[6] Dan, X., Wang, L., He, Z., 2024. Pdd: Pruning during distillation for efficient multi-agent reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[7] De Nijs, F., Walraven, E., De Weerdt, M., Spaan, M., 2021. Constrained multiagent markov decision processes: A taxonomy of problems and algorithms. Journal of Artificial Intelligence Research 70, 955–1001.

[8] Ellis, B., Cook, J., Moalla, S., Samvelyan, M., Sun, M., Mahajan, A., Foerster, J., Whiteson, S., 2023. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. Advances in Neural Information Processing Systems 36, 37567–37593.

[9] Foerster, J., Assael, I.A., De Freitas, N., Whiteson, S., 2016. Learning to communicate with deep multi-agent reinforcement learning. Advances in neural information processing systems 29.

[10] Gao, Y., Zhang, K., Yang, Y., Li, Y., Li, Z., Hu, H., 2021. Knowru: Knowledge reuse in multi-agent reinforcement learning. Neurocomputing 453, 464–475.

[11] Gou, J., Chen, Y., Yu, B., Liu, J., Du, L., Wan, S., Yi, Z., 2024. Reciprocal teacher-student learning via forward and feedback knowledge distillation. IEEE transactions on multimedia 26, 7901–7916.

[12] Gronauer, S., Diepold, K., 2022. Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review 55, 895–943.

[13] Harish, A.N., Heck, L., Hanna, J.P., Kira, Z., Szot, A., 2024. Reinforcement learning via auxiliary task distillation, in: European Conference on Computer Vision, Springer. pp. 214–230.

[14] Hu, C., Li, X., Liu, D., Wu, H., Chen, X., Wang, J., Liu, X., 2023. Teacher-student architecture for knowledge distillation: A survey. arXiv preprint arXiv:2308.04268 .

[15] Hu, P., Li, S., Li, Z., Pan, L., Huang, L., 2024. Value-based deep multi-agent reinforcement learning with dynamic sparse training. arXiv preprint arXiv:2409.19391 .

[16] Jiang, W., Feng, G., Qin, S., Liu, Y., 2019. Multi-agent reinforcement learning based cooperative content caching for mobile edge networks. IEEE Access 7, 61856–61867.

[17] Li, Z., Hu, X., Tang, J., 2025. Double distillation network for robust multi-agent coordination. IEEE Transactions on Pattern Analysis and Machine Intelligence In press.

[18] Li, Z., Xu, P., Dong, Z., Zhang, R., Deng, Z., 2024. Feature-level knowledge distillation for place recognition based on soft-hard labels teaching paradigm. IEEE Transactions on Intelligent Transportation Systems .

[19] Liu, D., Zhu, Y., Liu, Z., Liu, Y., Han, C., Tian, J., Li, R., Yi, W., 2025. A survey of model compression techniques: Past, present, and future. Frontiers in Robotics and AI 12, 1518965.

[20] Liu, K., Huang, Z., Wang, C.D., Gao, B., Chen, Y., 2024. Fine-grained learning behavior-oriented knowledge distillation for graph neural networks. IEEE Transactions on Neural Networks and Learning Systems .

[21] Liu, W., Chen, J., Zhang, M., 2023. Model compression in multi-agent reinforcement learning via reinforcement learning-guided pruning. IEEE Transactions on Neural Networks and Learning Systems To appear.

[22] Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems 30.

[23] Nekoei, H., Badrinaaraayanan, A., Sinha, A., Amini, M., Rajendran, J., Mahajan, A., Chandar, S., 2023. Dealing with non-stationarity in decentralized cooperative multi-agent deep reinforcement learning via multi-timescale learning, in: Conference on Lifelong Learning Agents, PMLR. pp. 376–398.

[24] Park, W., Kim, D., Lu, Y., Cho, M., 2019. Relational knowledge distillation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3967–3976.

[25] Pei, Y., Ren, T., Zhang, Y., Sun, Z., Champeyrol, M., 2025. Policy distillation for efficient decentralized execution in multi-agent reinforcement learning. Neurocomputing , 129617.

[26] Rashid, T., Samvelyan, M., De Witt, C.S., Farquhar, G., Foerster, J., Whiteson, S., 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. Journal of Machine Learning Research 21, 1–51.

[27] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., et al., 2018. Value-decomposition networks for cooperative multi-agent learning based on team reward, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2085–2087.

[28] Tseng, W.C., Wang, T.H.J., Lin, Y.C., Isola, P., 2022. Offline multi-agent reinforcement learning with knowledge distillation. Advances in Neural Information Processing Systems 35, 226–237.

[29] Wang, X., Zhao, Y., Liu, Q., 2022. Offline multi-agent reinforcement learning via knowledge distillation, in: Advances in Neural Information Processing Systems (NeurIPS).

[30] Wong, A., Bäck, T., Kononova, A.V., Plaat, A., 2023. Deep multiagent reinforcement learning: Challenges and directions. Artificial Intelligence Review 56, 5023–5056.

[31] Xu, Z., Wang, J., Xu, X., Yu, P., Huang, T., Yi, J., 2025. A survey of reinforcement learning-driven knowledge distillation: Techniques, challenges, and applications .

[32] Yang, C., Yu, X., Yang, H., An, Z., Yu, C., Huang, L., Xu, Y., 2025. Multi-teacher knowledge distillation with reinforcement learning for visual recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9148–9156.

[33] Yang, N., Chen, S., Zhang, H., Berry, R., 2024. Beyond the edge: An advanced exploration of reinforcement learning for mobile edge computing, its applications, and future research trajectories. IEEE Communications Surveys & Tutorials 27, 546–594.

[34] Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., Wu, Y., 2022. The surprising effectiveness of ppo in cooperative multi-agent games. Advances in neural information processing systems 35, 24611–24624.

[35] Zhang, R., Luo, Z., Sjölund, J., Schön, T., Mattsson, P., 2024. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. Advances in Neural Information Processing Systems 37, 98871–98897.

[36] Zhao, J., Hu, X., Yang, M., Zhou, W., Zhu, J., Li, H., 2022. Ctds: Centralized teacher with decentralized student for multiagent reinforcement learning. IEEE Transactions on Games 16, 140–150.

[37] Zhong, Y., Kuba, J.G., Feng, X., Hu, S., Ji, J., Yang, Y., 2024. Heterogeneous-agent reinforcement learning. Journal of Machine Learning Research 25, 1–67.

[38] Zhou, Y., Zheng, Y., Hu, Y., Chen, K., Zheng, T., Song, J., Song, M., Liu, S., 2025. Cooperative policy agreement: Learning diverse policy for offline marl, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 23018–23026.