PTDE: Personalized Training with Distilled Execution for Multi-Agent Reinforcement Learning

Yiqun Chen[1], **Hangyu Mao**[2], **Jiaxin Mao**[1*], **Shiguang Wu**[3], **Tianle Zhang**[4], **Bin Zhang**[5], **Wei Yang**[5], **Hongxing Chang**[5]
[1]Renmin University of China
[2]SenseTime
[3]Noah's Ark Lab, Huawei
[4]JD Explore Academy
[5]Institute of Automation,Chinese Academy of Sciences
chenyiqun990321@ruc.edu.cn, hy.mao@pku.edu.cn, {maojiaxin, weiyangvia}@gmail.com, wushiguang@huawei.com, tianle-zhang@outlook.com, {zhangbin2020, hongxing.chang}@ia.ac.cn

Abstract

Centralized Training with Decentralized Execution (CTDE) has emerged as a widely adopted paradigm in multi-agent reinforcement learning, emphasizing the utilization of global information for learning an enhanced joint Q-function or centralized critic. In contrast, our investigation delves into harnessing global information to directly enhance individual Q-functions or individual actors. Notably, we discover that applying identical global information universally across all agents proves insufficient for optimal performance. Consequently, we advocate for the customization of global information tailored to each agent, creating agent-personalized global information to bolster overall performance. Furthermore, we introduce a novel paradigm named Personalized Training with Distilled Execution (PTDE), wherein agent-personalized global information is distilled into the agent's local information. This distilled information is then utilized during decentralized execution, resulting in minimal performance degradation. PTDE can be seamlessly integrated with state-of-the-art algorithms, leading to notable performance enhancements across diverse benchmarks, including the SMAC benchmark, Google Research Football (GRF) benchmark, and Learning to Rank (LTR) task.

1 Introduction

Many real-world tasks can be modeled as decision problems for multi-agent systems, such as multi-robot navigation [Han *et al.*, 2020], multi-robot collision avoidance [Long *et al.*, 2018], multi-UAV path planning [Qie *et al.*, 2019], information retrieval [Chen *et al.*, 2024] and games [Mao *et al.*, 2021; Guss *et al.*, 2021]. In most of these scenarios, the prevalent constraints include partial observability, and agents are constrained to decentralized decision-making processes.

To address these challenges, Multi-Agent Reinforcement Learning (MARL) has emerged as a focal point of research. Within MARL, Centralized Training with Decentralized Execution (CTDE) stands out as a prominent paradigm. CTDE leverages global information during training and shifts to utilizing only local information during execution, facilitating decentralized decision-making. This paradigm encompasses two primary algorithmic categories: value-decomposition based and actor-critic based approaches. Concerning the utilization of global information, the former category [Sunehag *et al.*, 2017; Rashid *et al.*, 2018; Wang *et al.*, 2020; Chai *et al.*, 2021] employs global information for enhancing the joint Q-function. In contrast, the latter category [Lowe *et al.*, 2017; Foerster *et al.*, 2018; Iqbal and Sha, 2019; Yu *et al.*, 2021; Zhang *et al.*, 2023a] incorporates global information as input to a centralized critic. Notably, these approaches refrain from utilizing global information directly during execution, a factor that could potentially constrain collaboration performance among agents, especially in complex scenarios, as demonstrated in our experiments.

In contrast to the conventional CTDE approach, a distinct line of research explores the direct utilization of global information during execution. COPA [Liu *et al.*, 2021] introduces a Coach-Player framework, devising an adaptive communication method wherein the coach determines when to dispatch a global instruction vector to the players. This vector, combined with local information, is used to compute individual Q-functions. Despite COPA's use of a multi-head attention mechanism for comprehensive global information processing during execution, this information remains identical for all agents. In a different vein, the CSRL framework [Chen *et al.*, 2022] introduces a Commander-Soldiers MARL framework, incorporating the concept of specific information for each agent. Both COPA and CSRL notably enhance multi-agent collaboration performance by directly applying global information during execution.

Nevertheless, practical challenges emerge in numerous applications due to local observability constraints, posing difficulties in utilizing global information directly during execution. To reconcile the need for global information

*Corresponding author

while ensuring decentralized execution, we propose a novel paradigm named Personalized Training with Distilled Execution (PTDE), which comprises two training stages. In the first stage, we introduce the concept of *agent-personalized global information* by employing a Global Information Personalization (GIP) module. This module transforms raw global information into personalized global information tailored to each agent. This personalized global information is then utilized to compute individual Q-functions or individual policies, enhancing the performance of each agent. In the second stage, we implement knowledge distillation for the agent-personalized global information. Within this distillation framework, a proficiently trained GIP module acts as the teacher network, while a dedicated student network is employed for the distillation process. Crucially, the input for the student network is exclusively composed of the agent's local information, presenting a departure from the teacher network, which integrates both global and agent's local information [1].

During execution, the teacher network is replaced by the student network, enabling decentralized execution while retaining the benefits of personalized global information. This innovative approach ensures a seamless transition from personalized training to distilled execution within the proposed PTDE paradigm. Summary of our contributions:

- In contrast to the prevalent trend in CTDE-based methods, which emphasizes leveraging global information during centralized training, our approach shifts the focus to exploring the utilization of global information during decentralized execution.
- We identify that consistently positive performance among agents is challenging when applying the same global information for decision-making. However, our innovation lies in transforming global information into agent-personalized global information, resulting in agents consistently making improved decisions.
- We introduce a novel paradigm named PTDE, which not only benefit from agent-personalized global information but also executes in a decentralized manner through knowledge distillation. Importantly, our experiments demonstrate minimal performance degradation after distilling agent-personalized global information into agent's local information.
- Experimental results underscore the universality and efficacy of the PTDE paradigm across diverse multi-agent environments and algorithms.

2 Background

2.1 Dec-POMDP

In this work, we model a fully cooperative multi-agent task as the Dec-POMDP [Oliehoek and Amato, 2016], which is formally defined as a tuple $G = \langle \mathbf{S}, \mathbf{U}, P, r, \mathbf{Z}, \mathbf{O}, n, \gamma \rangle$. $s \in \mathbf{S}$ is the global state of the environment. Each agent $i \in \mathcal{A} \equiv \{1, ..., n\}$ chooses an action $u^i \in U$ which forms the joint action $\mathbf{u} \in \mathbf{U} \equiv U^n$. The state transition function is modeled as $P(s'|s, \mathbf{u}) : \mathbf{S} \times \mathbf{U} \times \mathbf{S} \to [0, 1]$. The reward function which is modeled as $r(s, \mathbf{u}) : \mathbf{S} \times \mathbf{U}$ is shared by all agents and the discount factor is $\gamma \in [0, 1)$. It follows partially observable settings, where agents do not have access to the global state. Instead, it samples observations $z \in \mathbf{Z}$ according to observation function $O(s, i) : \mathbf{S} \times \mathbf{U} \to \mathbf{Z}$. Each agent has an action-observation history trajectory $\tau^i \in T \equiv (\mathbf{Z} \times \mathbf{U})^*$, on which it conditions a stochastic policy $\pi^i(u^i|\tau^i) : T \times \mathbf{U} \to [0, 1]$. In our algorithm, the joint policy π is based on action-value function $Q^\pi(s_t, \mathbf{u}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{u}_{t+1:\infty}}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t, \mathbf{u}_t]$. The final goal is to get the optimal action-value function Q^*.

2.2 Typical MARL Algorithms

Value decomposition [Sunehag *et al.*, 2017; Rashid *et al.*, 2018; Wang *et al.*, 2020] and Actor-Critic [Yu *et al.*, 2021; Zhang *et al.*, 2022; Zhang *et al.*, 2023b; Hu *et al.*, 2024] are two typical branches of multi-agent reinforcement learning. Among these, VDN [Sunehag *et al.*, 2017] is the representative algorithm to formulate value-decomposition paradigm. QMIX [Rashid *et al.*, 2018] learns a monotonic factorisation ensuring that a global argmax operation on the joint action-value function Q_{tot} yield the same results as a series of individual argmax operations on each individual action-value function Q_i. QPLEX [Wang *et al.*, 2020] takes a duplex dueling network architecture to factorize the joint value function. MAPPO [Yu *et al.*, 2021] is an actor-critic based algorithm. To specialize for multi-agent settings, MAPPO uses the structure of PPO algorithm but the critic can take extra global information to follow the CTDE framework.

2.3 Knowledge Distillation

Knowledge distillation [Hinton *et al.*, 2015] is proposed to compress big models. It distills the knowledge generated from a larger network into a smaller network. Policy distillation [Rusu *et al.*, 2015] presents a novel knowledge distillation method which can be used in reinforcement learning to extract the policy of agent and train a new network with an expert level performance and better efficiency. CTDS [Zhao *et al.*, 2022] proposes a novel Centralized Teacher with Decentralized Student framework which consists of a teacher model and a student model to alleviate the inefficiency caused by the limitation of local observability.

3 Method

In this section, we initially present an approach that provides the same global information to all agents during execution for decision-making. Despite its simplicity, this naive use of global information does not consistently enhance multi-agent collaboration performance. Subsequently, we propose the Global Information Personalization (GIP) module to tailor global information for each agent, resulting in the agent-personalized global information. Based on this, we derive a centralized execution method that makes better use of global information for improved performance. Recognizing the challenge of directly obtaining global information during execution, we finally introduce the knowledge distillation ap-

[1] Our approach diverges from the conventional knowledge distillation employed in model compression [Gou *et al.*, 2021], where both the teacher and student networks operate on the same input.

proach to achieve decentralized execution without too much performance degradation.

3.1 Naive Use of Global Information

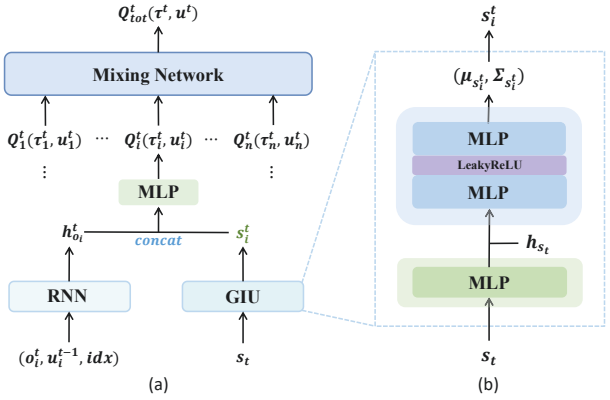


Figure 1: The framework of QMIX_GIU. (b) is the detail of the Global Information Unification (GIU) module.

We present a method that directly employs unified global information for all agents during execution. Using QMIX as an example, the entire framework is depicted in Figure 1 (a).

(1) The RNN module encodes the trajectory of the *agent's local information* $\mathbf{O_i^t} = (o_i^t, u_i^{t-1}, idx)$ to $h_{o_i}^t$.

(2) The Global Information Unification (GIU) module is designed to generate the *unified global information* s_i^t to be used in execution, where the green multilayer perceptron (MLP) encodes the *raw global information* s_t into h_{s_t} and the blue module transforms h_{s_t} into a multivariate gaussian distribution $\mathcal{N} \sim (\boldsymbol{\mu}_{\boldsymbol{s_i^t}}, \boldsymbol{\Sigma}_{\boldsymbol{s_i^t}})$.

(3) Similar to QMIX, the individual action-value $Q_i^t(\tau_i^t, u_i^t)$ is computed by an MLP operating on the concatenation of $h_{o_i}^t$ and s_i^t, while the joint action-value $Q_{tot}^t(\tau^t, u^t)$ is computed by nonlinearly combining all individual action-values through the mixing network.

Since the parameters of GIU module are unified and invariant to each agent during execution, the algorithm is called QMIX_GIU (**G**lobal **I**nformation **U**nification). It is one of the baselines and ablations in our experiments.

3.2 Global Information Personalization

In many multi-agent cooperative tasks, an agent's decision-making is significantly improved by concentrating on a subset of the global information, given that the raw global information tends to be redundant [Mao *et al.*, 2020]. Extracting and utilizing this relevant portion of global information is crucial for optimal decision-making. Inspired by this observation, we introduce the Global Information Personalization (GIP) module, crafted to autonomously tailor the global state (i.e., extracting the beneficial part) for each individual agent.

As shown in Figure 2, the GIP module comprises three components: Agent-Hyper Network, Agent-Personalization Network, and Distribution Generator. The Agent-Hyper Network takes the agent's local information as input and produces a set of weights W and biases B. The structures of

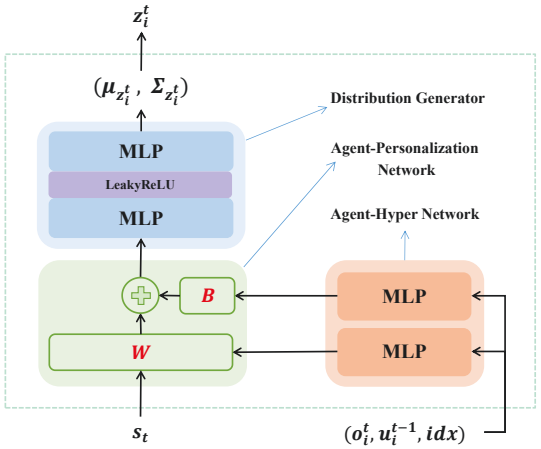


Figure 2: The structure of the Global Information Personalization (GIP) module.

the Agent-Personalization Network and Distribution Generator are identical to those in Figure 1 (b). The output of the GIP module, denoted as z_i^t, is defined by Equation (1).

$$z_i^t \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z_i^t}}, \boldsymbol{\Sigma}_{\boldsymbol{z_i^t}}) \tag{1}$$

$$\boldsymbol{\mu}_{\boldsymbol{z_i^t}} = f_\mu(\mathbf{O_i^t}, \mathrm{s}; \theta_\mu) \tag{2}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{z_i^t}} = f_\Sigma(\mathbf{O_i^t}, \mathrm{s}; \theta_\Sigma) \tag{3}$$

Compared to the GIU module (i.e., Figure 1 (b)), a distinguishing feature of GIP module is that the parameters of Agent-Personalization Network are dynamically generated by Agent-Hyper Network. Since the local information is different for each agent, the parameters of Agent-Personalization Network are guaranteed to be personalized to each agent, which is the key to global information personalization. Therefore, we call z_i^t the *agent-personalized global information* in this paper.

The GIP module is versatile across existing CTDE algorithms. In Figure 3 (a), the integration of the GIP module in value-decomposition methods is illustrated. Here, the individual Q-function is computed by concatenating $h_{o_i}^t$ and z_i^t. Figure 3 (b) showcases the application of the GIP module in actor-critic methods, where the concatenation $[h_{o_i}^t, z_i^t]$ is utilized by the individual actor for action sampling.

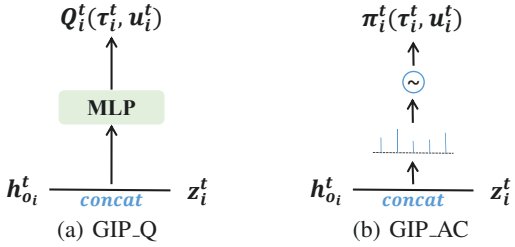


Figure 3: How GIP module is used in value-decomposition based methods (i.e., GIP_Q) and actor-critic based methods (i.e., GIP_AC).

3.3 Knowledge Distillation

The methods shown in Figure 3 involve the utilization of the global state s_t when computing individual Q-functions or individual policies (as z_i^t depends on s_t). However, obtaining global information directly is challenging due to partial observability in real-world multi-agent systems. To leverage global information during execution while adhering to the need for decentralized execution, we introduce a knowledge distillation method. This approach distills agent-personalized global information using only the agent's local information, i.e., transforming the dependence of z_i^t on s_t into the practical reliance on (o_i^t, u_i^{t-1}, idx).

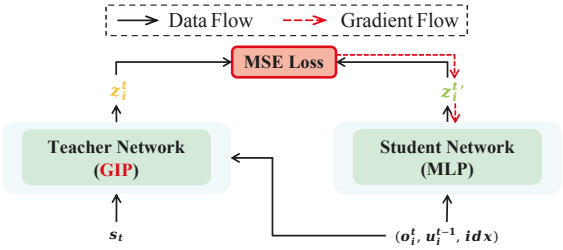


Figure 4: The knowledge distillation framework.

The knowledge distillation framework is illustrated in Figure 4. In this setup, the GIP module serves as the teacher network, while the student network is represented by an MLP. As indicated in Equation (4), the calculation of z_i^t involves s_t and $\mathbf{O_i^t} = (o_i^t, u_i^{t-1}, idx)$ through the teacher network. On the other hand, the student network's input consists solely of the agent's local information O_i^t, and its output is denoted as $z_i^{t'}$ in Equation (5). In the context of knowledge distillation, z_i^t is referred to as teacher knowledge, and $z_i^{t'}$ is student knowledge. Equation (6) illustrates the use of Mean Squared Error (MSE) Loss to minimize the disparity between student and teacher knowledge, ensuring effective training of the student network.

$$z_i^t = f_{tea}(\mathbf{O_i^t}, s_t) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{z_i^t}}(\mathbf{O_i^t}, s_t), \boldsymbol{\Sigma}_{\boldsymbol{z_i^t}}(\mathbf{O_i^t}, s_t)\right) \quad (4)$$

$$z_i^{t'} = f_{stu}(\mathbf{O_i^t}) = \mathbf{MLP}(\mathbf{O_i^t}) \quad (5)$$

$$\mathcal{L}_{mse} = ||z_i^{t'} - z_i^t||_2^2 = ||f_{tea}(\mathbf{O_i^t}, s_t) - f_{stu}(\mathbf{O_i^t})||_2^2 \quad (6)$$

Knowledge distillation is a common technique employed for model compression, typically involving identical input data for both teacher and student networks. In our knowledge distillation, a notable distinction arises as the student network's input lacks global information s_t compared to the teacher network. This divergence from traditional model compression is pivotal, serving as the key factor in transitioning from centralized execution to decentralized execution.

Upon completion of the knowledge distillation training, the student network can seamlessly replace the teacher network (i.e., the GIP module) during the execution process. In other words, $z_i^{t'}$ is utilized in lieu of z_i^t in Figure 3, enabling decentralized execution.

3.4 The Overall PTDE Paradigm

Figure 5 shows the overall framework of PTDE based on the QMIX algorithm, encompassing two-stage training and decentralized execution.

The First Training Stage. We provide personalized global information for each agent to compute a better individual Q-function or individual policy. Specifically, the RNN module records the trajectory of agent's local information and outputs an encoding vector $h_{o_i}^t$. The teacher network gives agent-personalized global information z_i^t. Then, the individual action-value $Q_i^t(\tau_i^t, u_i^t)$ can be computed by concatenation $[h_{o_i}^t, z_i^t]$. Finally, the joint action-value $Q_{tot}^t(\boldsymbol{\tau^t}, \boldsymbol{u^t})$ can be obtained by the mixing network. The whole network is trained end-to-end by minimizing the loss shown in Equation (7) and (8):

$$\mathcal{L}(\theta, \varphi) = \sum_{i=1}^{b} \left(y_i^{tot} - Q_{tot}(\boldsymbol{\tau}, \mathbf{u}, s; \theta, \varphi)\right)^2 \quad (7)$$

$$y^{tot} = r + \gamma \max_{u'} Q_{tot}(\boldsymbol{\tau}', \mathbf{u}', s'; \theta^-, \varphi^-) \quad (8)$$

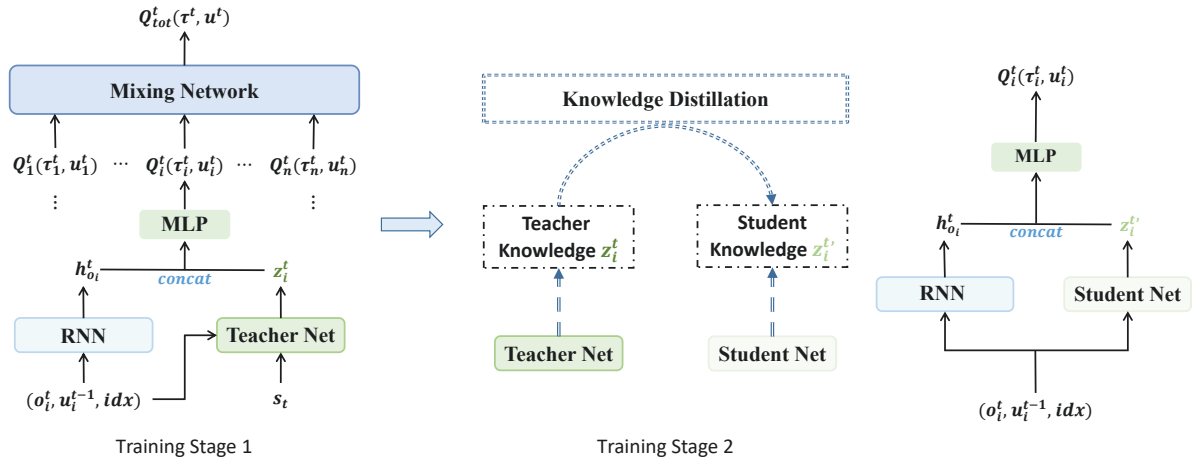
where b is the batch size of sampled experiences from replay buffer; φ is the parameters of the teacher network and θ is parameters of the other networks in the first training stage; θ^- and φ^- are parameters of target networks. We also show the pseudo-code in **Algorithm 1**.

Algorithm 1: The first training stage of PTDE

Initialize: The parameters θ and φ of network, θ^- and φ^- of target network, replay buffer $\mathcal{D}$.
Initialize: Observation $\boldsymbol{o} = (o_1, \cdots, o_N)$ and state $\boldsymbol{s}$.
while *not over* **do**
  Collect a tuple $(\boldsymbol{o}, \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{o'}, \boldsymbol{s'})$ by generating 8 parallel episodes, and store it in $\mathcal{D}$;
  Sample a random minibatch $\boldsymbol{b}$ from $\mathcal{D}$;
  The RNN calculate $h_{o_i}^t$, the teacher network calculate z_i^t;
  Calculate y^{tot} and loss $\mathcal{L}$ for all sampled data from $\boldsymbol{b}$ based on Equation (8) and (7);
  Update the parameters of networks θ, φ;
  Update the parameters of target networks θ^-, φ^- every N episodes;
Output: Get a well-trained teacher network and an algorithm that works well to execute centrally.

The Second Training Stage. After the first training stage, the teacher network can provide agent-personalized global knowledge z_i^t for agents' decision-making. In the second stage, we use offline knowledge distillation where the student network distills teacher knowledge z_i^t to obtain student knowledge $z_i^{t'}$. Subsequently, the student network can replace the teacher network during the execution process. The pseudo-code can be seen in **Algorithm 2**.

Decentralized Execution. As shown in Figure 5 (b), agents utilize solely local information (o_i^t, u_i^{t-1}, idx) to compute individual action-values $Q_i^t(\tau_i^t, u_i^t)$ and sample actions using $u_i^t = \arg\max_u Q_i^t(\tau_i^t, u_i^t)$. We name this method as QMIX_KD.



(a) Two-stage training (b) Decentralized Execution

Figure 5: The framework of PTDE: Two-Stage Training and Decentralized Execution.

Algorithm 2: The second training stage of PTDE

Initialize: The student network parameters ψ.
Load Model: Load the models and parameters θ and φ in Algorithm 1.
Generate Data: Generate 100 episodes data, including s_t and (o_i^t, u_i^{t-1}, idx), and save offline.
Offline Train: Use MSE Loss in Equation (6) to train the student network offline for multiple epochs.
Output: Get a well-trained student network.

	Algorithm	Description
Centralized Execution	CSRL	[Chen *et al.*, 2022]
	COPA	[Liu *et al.*, 2021]
	QMIX_GIU	Unified global information is used during execution. (Proposed in Section 3.1)
	QMIX_GIP (stage1)	Agent-Personalized global information is used during execution.
Decentralized Execution	QMIX_KD (stage2)	Student knowledge (rather than teacher knowledge) is used during execution.
	QMIX	[Rashid *et al.*, 2018]
	QPLEX	[Wang *et al.*, 2020]

Table 1: Algorithms and baselines in experiments.

4 Experiments

Our experiments mainly focus on six research questions:

(**RQ.1**) Does the utilization of unified global information during execution lead to an enhancement in the performance of multi-agent collaboration?

(**RQ.2**) Is agent-personalized global information more effective in improving performance compared to unified global information?

(**RQ.3**) After knowledge distillation, can the algorithm maintain a substantial portion of its performance when transitioning from centralized to decentralized execution?

(**RQ.4**) Does the PTDE-based algorithm exhibit universality across diverse environments?

(**RQ.5**) Does the PTDE paradigm demonstrate universality across various algorithms?

(**RQ.6**) What is the rationale behind the PTDE paradigm's approach of partitioning the training process into two stages?

We investigate the research questions using popular MARL testbeds, namely StarCraft II [Samvelyan *et al.*, 2019] and Google Research Football [Kurach *et al.*, 2020]. Additionally, we validate RQ.4 through experiments in the Learning to Rank (LTR) scenario [Liu and others, 2009]. To our best knowledge, this is the first time that the MARL algorithm has been applied to LTR tasks.

For baselines, we categorize them into two classes: centralized execution algorithms and decentralized execution algorithms, outlined in Table 1. To showcase the impact of agent-personalized global information, we contrast our approach with two centralized execution algorithms, CSRL and COPA, and perform an ablation experiment using QMIX_GIU. Furthermore, to assess algorithm performance after knowledge distillation, we utilize QMIX and QPLEX as decentralized execution baselines. All experiments are conducted using the PyMARL2 framework [Hu *et al.*, 2021] with 8 parallel runners and 3 random seeds. Details regarding hyperparameters are available in Table 7 in the Appendix.

4.1 StarCraft II

To address **RQ.1**, **RQ.2**, and **RQ.3**, we select hard scenarios such as *5m_vs_6m* and *3s_vs_5z*, as well as super hard scenarios like *3s5z_vs_3s6z* and *6h_vs_8z*. Additionally, to further highlight the advantages of agent-personalized global information in multi-agent collaboration, we conduct experiments in two new scenarios, namely *3s_vs_8z* (featuring 8 zealots in the enemy team) and *3s5z_vs_3s7z* (with 3 stalkers and 7 zealots in the enemy team), where existing decentralized execution algorithms exhibit poor performance.

As shown in Table 2, QMIX_GIU has better performance than QMIX in *3s_vs_5z*, *5m_vs_6m* and *6h_vs_8z*, but performs worse in *3s_vs_8z* and *3s5z_vs_3s6z*. This addresses **RQ.1**, indicating that unified global information can have a positive impact on decision-making in certain scenarios but

	Algorithms	*3s_vs_5z* (2M)	*5m_vs_6m* (2M)	*3s5z_vs_3s6z* (5M)	*6h_vs_8z* (5M)	*3s_vs_8z* (10M)	*3s5z_vs_3s7z* (10M)
Centralized Execution	CSRL	0.917±0.031	0.733±0.063	0.396±0.267	0.502±0.373	0.934±0.019	0.109±0.092
	COPA	0.748±0.087	0.688±0.117	0.064±0.098	0.709±0.129	0.490±0.230	0.000±0.000
	QMIX_GIU	0.868±0.095	0.696±0.079	0.026±0.040	0.405±0.310	0.332±0.264	0.000±0.000
	QMIX_GIP (stage1)	**0.992±0.006**	**0.806±0.008**	**0.776±0.062**	**0.712±0.053**	**0.990±0.002**	**0.710±0.152**
Decentralized Execution	**QMIX_KD (stage2)**	**0.887±0.027**	**0.690±0.088**	**0.674±0.069**	**0.524±0.085**	**0.576±0.055**	**0.631±0.201**
	QMIX	0.128±0.165	0.586±0.068	0.140±0.081	0.012±0.019	0.355±0.197	0.000±0.000
	QPLEX	0.000±0.000	0.616±0.070	0.390±0.145	0.021±0.034	0.074±0.066	0.000±0.000
Performance Retention Ratio (PRR)		89.4%	85.6%	86.9%	73.6%	58.2%	88.9%

Table 2: Winning rates on StarCraft II.

may not consistently improve agent decisions. In contrast, QMIX_GIP consistently outperforms QMIX_GIU in all testing scenarios, supporting **RQ.2** by highlighting the consistent benefits of agent-personalized global information. Furthermore, QMIX_GIP attains the highest winning rates among all centralized execution algorithms, further affirming the advantages of agent-personalized global information (**RQ.2**).

The Performance Retention Ratio (PRR) in Table 2 signifies the ratio of the winning rates of QMIX_KD to those of QMIX_GIP, reflecting the performance before and after knowledge distillation. The PRRs range between 85% and 90% in four out of six simulation maps, indicating that the PTDE paradigm can maintain performance reasonably well after knowledge distillation (**RQ.3**). Notably, even in challenging scenarios like *3s5z_vs_3s7z*, where all baselines have low winning rates, QMIX_KD achieves a winning rate of 63.1%, showcasing the substantial advantages of PTDE in such extreme conditions. Overall, the PTDE paradigm's ability to train a viable strategy with the assistance of agent-personalized global information and subsequently achieve decentralized execution through knowledge distillation is highlighted. The training curves (Figure 9 in Appendix C) and strategy visualizations (Appendix A) also illustrate the performance improvement of QMIX_KD over QMIX.

4.2 Google Research Football

To further validate **RQ.1**, **RQ.2**, and **RQ.3**, we select five widely recognized academy scenarios: *3_vs_1_with_keeper*, *3_vs_2_with_keeper*, *counterattack_easy*, *counterattack_hard* and *run_pass_and_shoot_with_keeper*. The agents are trained for 10 million steps using 8 threads in all scenarios.

Table 3 displays winning rates on GRF. QMIX_GIP achieves the best performance across all scenarios, reinforcing the notion that agent-personalized global information is more beneficial for multi-agent collaboration than unified global information (**RQ.2**). The high PRRs further demonstrate that the performance does not degrade significantly after knowledge distillation (**RQ.3**). Specifically, in *3_vs_1_with_keeper* and *run_pass_and_shoot_with_keeper*, PRRs range from 95% to 100%, while in *counterattack_easy* and *counterattack_hard*, PRRs range from 80% to 90%. These conclusions align with the training curves available in Figure 10 in Appendix C.

4.3 Scenario Universality of PTDE Paradigm

To further validate the effectiveness of the PTDE paradigm, we extend its application to the Learning to Rank (LTR) [Liu and others, 2009] task. Ranking plays a crucial role in information retrieval, where the goal is to arrange a list of candidate documents in descending order of relevance to a given query. Achieving an optimal search ranking list enhances the effectiveness of information retrieval.

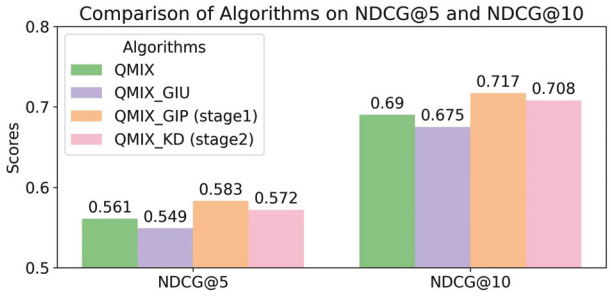


Figure 6: Experiments on Learning to Rank task.

In the multi-agent cooperation setting for the Learning to Rank (LTR) task, each document is treated as an agent. The fundamental components of Multi-Agent Reinforcement Learning (MARL) are defined as follows:

- Observation: Features of the query and document i.
- State: Features of the query and all documents.
- Reward: Given that NDCG@k [Järvelin and Kekäläinen, 2002] is a standard evaluation metric for ranking, we define the reward as NDCG@k.
- Action Space: Discrete scores, such as integers from 0 to 9, where the document is ultimately sorted based on the scores assigned to each document (agent).

We conducted training and testing on 10,000 queries (7:3 partition) from the MSLR-WEB30K [Qin and Liu, 2013] dataset, a large-scale dataset for Learning to Rank research. We modified this dataset to comprise 10 documents per query, resulting in a total of 10,000 queries and 100,000 documents. The experimental results are depicted in Figure 6. Our approaches, QMIX_GIP and QMIX_KD, achieve higher NDCG scores compared to QMIX and QMIX_GIU, addressing **RQ.2** and **RQ.3**. Interestingly, QMIX_GIU performs worse than QMIX, providing additional insights into **RQ.1**.

Crucially, the experiments in LTR, along with those on the StarCraft II and GRF benchmarks, collectively demonstrate that the PTDE paradigm exhibits good universality across diverse scenarios, providing a response to **RQ.4**.

4.4 Algorithm Universality of PTDE Paradigm

In this section, we integrate the PTDE paradigm with VDN and MAPPO, and test them on the *3s_vs_5z* and *3s5z_vs_3s6z*

	Algorithms	*3_vs_1_w_keeper*	*3_vs_2_w_keeper*	*counterattack_easy*	*counterattack_hard*	*run_pass_and_shoot_w_keeper*
Centralized Execution	QMIX_GIU	0.662±0.256	0.415±0.201	0.839±0.075	0.462±0.240	0.687±0.072
	QMIX_GIP (stage1)	**0.858±0.032**	**0.664±0.209**	**0.839±0.035**	**0.636±0.074**	**0.779±0.082**
Decentralized Execution	**QMIX_KD (stage2)**	**0.818±0.056**	**0.732±0.138**	**0.734±0.176**	**0.517±0.053**	**0.775±0.055**
	QMIX	0.609±0.250	0.491±0.200	0.365±0.165	0.184±0.174	0.533±0.201
Performance Retention Ratios (PRR)		95.3%	110.2%	87.5%	81.3%	99.5%

Table 3: Winning rates on Google Research Football.

Algorithm	3s_vs_5z (5M)	3s5z_vs_3s6z (10M)
VDN	0.653	0.397
VDN_GIU	0.971	0.155
VDN_GIP (stage1)	0.990	0.704
VDN_KD (stage2)	0.889	0.609

Table 4: Apply PTDE paradigm to VDN.

Algorithm	3s_vs_5z (5M)	3s5z_vs_3s6z (10M)
MAPPO	0.767	0.000
MAPPO_GIU	0.602	0.486
MAPPO_GIP (stage1)	0.965	0.694
MAPPO_KD (stage2)	0.891	0.589

Table 5: Apply PTDE paradigm to MAPPO.

scenarios. We adopt hyperparameters as specified in [Hu *et al.*, 2021] and [Yu *et al.*, 2021], respectively. As shown in Table 4 and 5, VDN_GIU's performance on *3s5z_vs_3s6z* is inferior to that of VDN, and MAPPO_GIU's performance on *3s_vs_5z* is worse than MAPPO. This highlights that unified global information does not always contribute to the efficacy of multi-agent collaboration (**RQ.1**). Furthermore, VDN_GIP outperforms VDN_GIU, and MAPPO_GIP outperforms MAPPO_GIU in both scenarios, demonstrating the effectiveness of the PTDE paradigm for both value-decomposition-based algorithm VDN and actor-critic-based algorithm MAPPO. In essence, the PTDE paradigm exhibits good universality across different algorithm types (**RQ.5**). The training curves for Table 4 and Table 5 can be found in Figure 11 in Appendix C.

4.5 Empirical Analysis of the Two-Stage Training

Why is it necessary to divide the training process into two stages? In Table 6, we compare the PTDE and CTDS paradigms based on two metrics: winning rate and PRR. CTDS synchronizes the distillation of global policies with centralized training, while our PTDE approach first conducts centralized training and then performs agent-personalized global knowledge distillation. Across the *5m_vs_6m*, *6h_vs_8z*, and *3s5z_vs_3s7z* maps, the PRR metric consistently favors the PTDE paradigm over the CTDS paradigm. This underscores that the two-stage training approach of PTDE maintains superior performance during decentralized execution (addressing **RQ.3** and **RQ.6**).

Figure 7 displays the loss function (6) and test winning rates during the knowledge distillation process on the *3s_vs_5z* scenario. The loss plot displays training and testing loss curves spanning 0 to 40k steps. The winning rate plot showcases a blue line representing test winning rates for decentralized execution after knowledge distillation, and a red dashed line signifying the test winning rate for centralized execution before knowledge distillation. In this experiment,

Algorithm	5m_vs_6m	6h_vs_8z	3s5z_vs_3s7z
QMIX_GIP (stage1)	0.806	0.712	0.710
QMIX_KD (stage2)	0.690	0.524	0.631
PRR	85.6%	73.6%	88.9%
CTDS (QMIX_Teacher)	0.698	0.367	0.000
CTDS (QMIX_Student)	0.490	0.204	0.000
PRR	70.2%	55.6%	-

Table 6: Comparisons between PTDE and CTDS.

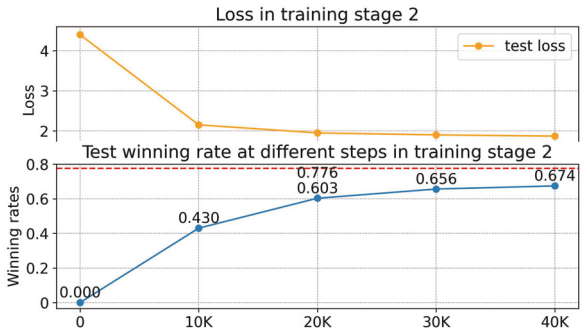


Figure 7: The loss and winning rates in training stage 2.

the batch size is set to 1000, and each step corresponds to the training of one batch. As can be observed, a notable reduction in loss occurs before 30k steps, accompanied by a swift increase in the test winning rate. This suggests that distilling global policies using local information requires a specific training duration. Conversely, simultaneous knowledge distillation during centralized training might fail to preserve optimal performance in decentralized execution due to inadequate training and uncertainties in the distribution of training samples. This analysis addresses and responds to **RQ.6**.

5 Conclusions

We introduced a two-stage training paradigm, named Personalized Training with Distilled Execution (PTDE), designed for multi-agent reinforcement learning. In the first training stage, the Global Information Personalization (GIP) module tailors global information for each agent. Subsequently, during the second training stage, the student network distills agent-personalized global information using solely the local information of each agent. In the execution stage, the student network takes over from the teacher network, enabling decentralized execution.

Our empirical evaluations on the StarCraft II benchmark, Google Research Football benchmark, and Learning to Rank task collectively offer conclusive answers to the posed research questions (RQ.1 to RQ.6). These results provide robust evidence supporting the efficacy and broad applicability of the PTDE paradigm.

Acknowledgments

This research was supported by the Natural Science Foundation of China (61902209, 62377044, U2001212), and Beijing Outstanding Young Scientist Program (NO.BJJWZYJH012019100020098) and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

[Chai *et al.*, 2021] Jiajun Chai, Weifan Li, Yuanheng Zhu, Dongbin Zhao, Zhe Ma, Kewu Sun, and Jishiyu Ding. Unmas: Multiagent reinforcement learning for unshaped cooperative scenarios. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2021.

[Chen *et al.*, 2022] Yiqun Chen, Wei Yang, Tianle Zhang, Shiguang Wu, and Hongxing Chang. Commander-soldiers reinforcement learning for cooperative multi-agent systems. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022.

[Chen *et al.*, 2024] Yiqun Chen, Jiaxin Mao, Yi Zhang, Dehong Ma, Long Xia, Jun Fan, Daiting Shi, Zhicong Cheng, and Dawei Yin. Ma4div: Multi-agent reinforcement learning for search result diversification. *arXiv preprint arXiv:2403.17421*, 2024.

[Foerster *et al.*, 2018] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Gou *et al.*, 2021] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[Guss *et al.*, 2021] William Hebgen Guss, Stephanie Milani, Nicholay Topin, Brandon Houghton, Sharada Mohanty, Andrew Melnik, Augustin Harter, Benoit Buschmaas, Bjarne Jaster, Christoph Berganski, Hangyu Mao, et al. Towards robust and domain agnostic reinforcement learning competitions: Minerl 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 233–252. PMLR, 2021.

[Han *et al.*, 2020] Ruihua Han, Shengduo Chen, and Qi Hao. Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 448–454. IEEE, 2020.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[Hu *et al.*, 2021] Jian Hu, Siyang Jiang, Seth Austin Harding, Haibin Wu, and Shih-wei Liao. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv e-prints*, pages arXiv–2102, 2021.

[Hu *et al.*, 2024] Tianyi Hu, Zhiqiang Pu, Xiaolin Ai, Tenghai Qiu, and Jianqiang Yi. Measuring policy distance for multi-agent reinforcement learning. *arXiv preprint arXiv:2401.11257*, 2024.

[Iqbal and Sha, 2019] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pages 2961–2970. PMLR, 2019.

[Järvelin and Kekäläinen, 2002] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[Kurach *et al.*, 2020] Karol Kurach, Anton Raichuk, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4501–4510, 2020.

[Liu and others, 2009] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[Liu *et al.*, 2021] Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*, pages 6860–6870. PMLR, 2021.

[Long *et al.*, 2018] Pinxin Long, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6252–6259. IEEE, 2018.

[Lowe *et al.*, 2017] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[Mao *et al.*, 2020] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning agent communication under limited bandwidth by message pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5142–5149, 2020.

[Mao *et al.*, 2021] Hangyu Mao, Chao Wang, Xiaotian Hao, Yihuan Mao, Yiming Lu, Chengjie Wu, Jianye Hao, Dong Li, and Pingzhong Tang. Seihai: A sample-efficient hierarchical ai for the minerl competition. In *International Conference on Distributed Artificial Intelligence*, pages 38–51. Springer, 2021.

[Oliehoek and Amato, 2016] Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

[Qie *et al.*, 2019] Han Qie, Dianxi Shi, Tianlong Shen, Xinhai Xu, Yuan Li, and Liujing Wang. Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. *IEEE access*, 7:146264–146272, 2019.

[Qin and Liu, 2013] Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 4295–4304. PMLR, 2018.

[Rusu *et al.*, 2015] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[Samvelyan *et al.*, 2019] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

[Sunehag *et al.*, 2017] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

[Wang *et al.*, 2020] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.

[Yu *et al.*, 2021] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

[Zhang *et al.*, 2022] Bin Zhang, Yunpeng Bai, Zhiwei Xu, Dapeng Li, and Guoliang Fan. Efficient policy generation in multi-agent systems via hypergraph neural network. In *International Conference on Neural Information Processing*, pages 219–230. Springer, 2022.

[Zhang *et al.*, 2023a] Bin Zhang, Lijuan Li, Zhiwei Xu, Dapeng Li, and Guoliang Fan. Inducing stackelberg equilibrium through spatio-temporal sequential decision-making in multi-agent reinforcement learning, 2023.

[Zhang *et al.*, 2023b] Bin Zhang, Hangyu Mao, Lijuan Li, Zhiwei Xu, Dapeng Li, Rui Zhao, and Guoliang Fan. Stackelberg decision transformer for asynchronous action coordination in multi-agent systems. *arXiv preprint arXiv:2305.07856*, 2023.

[Zhao *et al.*, 2022] Jian Zhao, Xunhan Hu, Mingyu Yang, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. Ctds: Centralized teacher with decentralized student for multi-agent reinforcement learning. *arXiv preprint arXiv:2203.08412*, 2022.