Meta Learning Task Representation in Multiagent Reinforcement Learning: From Global Inference to Local Inference

Zijie Zhao, *Graduate Student Member, IEEE*, Yuqian Fu, *Graduate Student Member, IEEE*, Jiajun Chai, *Graduate Student Member, IEEE*, Yuanheng Zhu, *Senior Member, IEEE*, and Dongbin Zhao, *Fellow, IEEE*

***Abstract*— Multiagent meta reinforcement learning (MAMRL) enables multiagent systems (MASs) to adapt to multiple tasks. However, partial observability poses a significant challenge by hindering efficient task inference from agents' limited local experiences. To address this, we propose MG2L, a novel algorithm featuring a global-to-local (G2L) training scheme based on mutual information optimization (MIO). We first extend the centralized training and decentralized execution (CTDE) framework to MAMRL, and introduce a multilevel task encoder for joint global and local task inference. Building on this encoder, the MG2L scheme employs tailored loss functions to optimize task representations. For global inference, the MAS learns a centralized global representation by maximizing the MI between the representation and the task context. For local inference, we formulate conditional MI reduction to quantify the G2L gap. Agents then learn the local representation by minimizing this reduction. The MG2L scheme effectively harmonizes centralized training with decentralized execution, offering a versatile solution for MAMRL challenges. Additionally, we integrate a permutation-invariant attention (PIA) module into the task encoder to reduce sensitivity to behavior policy variations. Extensive experiments—including comparative analyses, ablation studies, meta-test evaluations, and visualizations—demonstrate MG2L's effectiveness. The implementation of MG2L is publicly available at https://github.com/zhaozijie2022/mg2l.**

***Index Terms*— Meta-reinforcement learning (meta-RL), multiagent reinforcement learning (MARL), mutual information optimization (MIO), task inference.**

I. Introduction

MULTIAGENT reinforcement learning (MARL) has demonstrated notable success in cooperative applications ranging from game theory [1], video games [2], [3], and multirobot communication [4] to multipath planning [5]. As multiagent systems (MASs) [6] are increasingly applied in real-world scenarios, there is a growing demand for systems that can adapt to dynamic environments while simultaneously performing multiple tasks. This has spurred research into enhancing adaptability for novel tasks [7] and few-shot generalization [8], with meta-reinforcement learning (meta-RL) emerging as a key methodology.

Meta-RL trains agents on a distribution of tasks, allowing them to acquire shared knowledge from similar tasks and enabling rapid adaptation to new tasks. Meta-RL broadly falls into two categories: gradient-based approaches [9] and context-based approaches [10], [11]. In gradient-based approaches, agents learn an optimized initial model, and adapt to multiple tasks through online rollouts and gradient descent iterations. On the other hand, in context-based methods, agents treat their interaction historical experience with the environment as context, learning task representations from this context, and establishing policies conditioned on these representations. Context-based methods have the advantage of not requiring online fine-tuning for new tasks, making them well-suited for the multiagent setting, particularly in distributed MASs.

Nonetheless, there are still several challenges in extending existing context-based meta-RL methods to the multiagent setting. First, due to the influence of other agents' actions, an individual agent faces a nonstationarity environment. This means that even if local observations and actions are the same, state transitions may vary. Therefore, the common auxiliary task in context-based methods, predicting the successor state [12], is no longer applicable. Additionally, standalone RL losses often exhibit stochasticity and high variance. Thus, new losses need to be introduced. Second, the challenge of partial observability becomes notably prominent in the meta setting. In distributed MASs, the local experience of an individual agent may only encompass a fraction of the task-specific features, and the local task inference may not yield efficient task representations. To tackle the partial observability challenge, centralized training and decentralized execution (CTDE) framework [13] proves to be a suitable choice. As illustrated in Fig. 1, meta-CTDE introduces global task inference during centralized meta-training, constructing a global-to-local (G2L) structure.

Based on the above framework of meta-CTDE, we present MG2L, a novel multiagent meta reinforcement learning

Received 23 January 2024; revised 25 October 2024; accepted 4 February 2025. This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030400, in part by the National Natural Science Foundation of China under Grant 62293541 and Grant 62136008, in part by the Beijing Natural Science Foundation under Grant 4232056, and in part by the Beijing Nova Program under Grant 20240484514. *(Corresponding authors: Yuanheng Zhu; Dongbin Zhao.)*

The authors are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhaozijie22@mails.ucas.ac.cn; yuanheng.zhu@ia.ac.cn; dongbin.zhao@ia.ac.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3540758

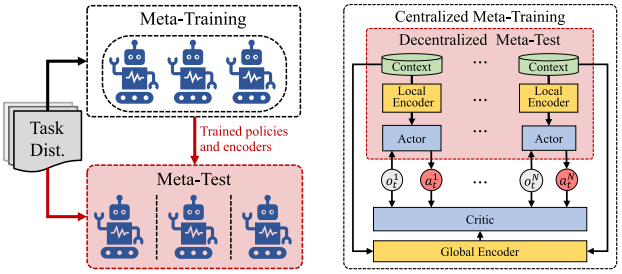

Fig. 1. Framework of (a) MAMRL and (b) meta-CTDE.

(MAMRL) algorithm with a **M**utual information optimization-based **g**lobal-**to**-**l**ocal training scheme. The contributions of this article can be summarized as follows.

1) Inspired by the CTDE framework [13], we initially formulate meta-CTDE to extend single-agent meta-RL to a multiagent setting. The multilevel task encoder is proposed to facilitate global and local inference. During centralized meta-training, global inference utilizes the context of all agents, while decentralized meta-test involves local inference relying only on local context. The tiered and highly modular structure of the encoder lays the foundation for subsequent training schemes and loss functions.
2) Building upon mutual information optimization (MIO), we present the MG2L training scheme with associated loss functions. For global inference, Info Noise Contrastive Estimation (InfoNCE) [14] is employed to maximize MI between the global representation and the task. For local inference, we maximize the conditional MI to enhance the utilization of global context and thereby reduce the gap from G2L. In the MG2L scheme, the centralized global representation is trained to encompass as rich task-specific features, and individual agents learn local representations aligned the global representation from the aspect of MI.
3) We integrate a permutation-invariant attention (PIA) module into the multilevel task encoder. The attention mechanism enables agents to quickly focus on and extract task-specific features while filtering out task-redundant features. The permutation-invariance reduces the impact of the changes in behavior policy, enhancing training stability.
4) We validate the effectiveness of MG2L through diverse comparative experiments, visualize the generalization process of MG2L, conduct ablation studies to analyze the impact of different components on performance.

The structure of the remaining sections is outlined as follows: Section II reviews the related work. Section III introduces the preliminaries of context-based MAMRL. Section IV details the proposed method MG2L and provides the theoretical analysis. Section V presents the experiments and results. Finally, Section VI concludes this article.

II. Related Work

A. Meta Reinforcement Learning

The core idea of meta-RL is to train an agent to rapidly adapt to multiple similar tasks. Related works in this domain can be broadly classified into two main categories: gradient-based approaches and context-based approaches. Gradient-based approaches, as exemplified by model-agnostic meta-learning (MAML) [9] revolve around the idea of learning an optimized initial model to maximize performance on new tasks with minimal rollouts and gradient descent iterations. On the other hand, context-based approaches aim to learn a latent task representation from historical experience. The agent can rapidly adapt to new tasks by inferring the latent representation from the context of the task at hand. For instance, RL^2 [10] employs a recurrent neural network (RNN) to extract task-specific features from trajectories, utilizing the hidden layer state in the network to infer the task. Probabilistic embeddings for actor-critic RL (PEARL) [11] employs a variational autoencoder (VAE) to infer tasks from transition tuples, utilizing the Gaussian product to aggregate the distribution of latent representations.

In contrast to gradient-based approaches, context-based approaches do not require updating the network during adaptation. This characteristic makes them more widely applicable, especially in distributed MASs.

B. Multitask MARL and MAMRL

Following the paradigm of CTDE [13], MARL has made significant progress in solving cooperative problems [15]. There are also many studies focusing on multitask MARL and MAMRL. For the multitask setting, unshaped networks for multiagent systems (UNMAS) [7] proposes self-weighting mixing to adapt to variations in the number of agents. Offline MARL algorithm to discover coordInation skills (ODIS) [16] learns task-invariant coordination skills from offline data to enhance team collaboration. For the MAMRL setting, MAS is regarded as a whole meta-learner, where the system leverages historical experiences to accelerate the generalization to new tasks [5], [17]. For instance, multi-agent task embeddings (MATE) [18] employs an encoder–decoder structure to learn multiagent task embeddings, which are used to improve team coordination. However, current research primarily concentrates on team collaboration and diverse team sizes, with limited emphasis on the adaptation of MASs to varying task content and environments. The primary challenge in this lies in the heightened sensitivity of algorithms to task variations, due to the partial observability of MASs [19], [20]. In this article, we will present a G2L solution based on MIO.

C. Mutual Information and Contrastive Learning

MI, a basic concept describing the dependence between two random variables (RVs), is frequently employed in designing objective functions [21]. Consequently, numerous studies concentrate on estimating the upper and lower bounds of MI. For the lower bound, InfoNCE [14] provides a lower bound I_{NCE} by separating the similarity of positive and negative samples. Conversely, with respect to the upper bound, Alemi et al. [22] introduce a variational upper bound by incorporating an auxiliary variational distribution. Poole et al. [23] replace the auxiliary variational distribution with a Monte Carlo approximation, and derives a new upper bound, termed as leave-one-out bound I_{L1o}. Due to their similar structures and independence from additional networks and parameters, the

lower bound I_{NCE} and upper bound I_{L1o} find widespread applications in various representation learning issues.

Contrastive learning (CL), a popular branch of self-supervised representation learning, is extensively applied in computer vision [24] and data knowledge [25]. There are also several works focusing on the fusion of CL with context-based meta-RL: fully-offline context-based actor-critic meta-RL (FOCAL) [26] designs a task encoder trained by distance metric learning, contrastive learning augmented context-based meta-RL (CCM) [27] pioneers CL and MI optimize in context-based Meta-RL, and contrastive robust task representation learning for OMRL (CORRO) [28] presents a method for generating negative samples in offline settings. However, to the best of our knowledge, due to the distributed challenges in MASs, there is currently no work focusing on the combination CL and MAMRL.

III. Preliminaries

A. Cooperative Multiagent Reinforcement Learning

A co-operative multiagent game can be formulated as a decentralized partially observable Markov decision process (dec-POMDP) [13]. A dec-POMDP is defined by a tuple $\langle \mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \Omega, P, R, \gamma \rangle$, where $\mathcal{I} = \{1, \ldots, n\}$ is the indices of agents, $\mathcal{S}$ denotes the state space, $\mathcal{O} = \prod_{i=1}^{n} O^i$ denotes the joint observation space, $\mathcal{A} = \prod_{i=1}^{n} A^i$ denotes the joint action space, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longmapsto [0, 1]$ denotes the transition function that returns a distribution of successor state conditioned on the current state and the joint action. At timestep t, each agent i perceives a local observation $o_t^i \in \mathcal{O}^i$ through the observation function $\Omega(s_t, i)$, and takes an action $a_t^i \in \mathcal{A}^i$ according to its local policy $\pi^i(a_t^i|o_t^i)$. After the joint action $\boldsymbol{a_t} = (a_t^1, \ldots, a_t^n)$ is executed, the successor state s_{t+1} is sampled from the distribution $P(s_{t+1}|s_t, \boldsymbol{a_t})$, and the reward r_t^i is given by the reward function $R(s_t, \boldsymbol{a_t})$, specifically, the agents share the same reward r_t in cooperative tasks. Based on the above protocol, the objective of a cooperative MAS is to optimize the joint policy to maximize the discounted accumulated reward $G_t = \sum_{t=0}^{\infty} \gamma^t r_t$, where $\gamma \in [0, 1)$ is the discount factor. To this end, the value function is defined to represent the value of the current state

$$V^{\boldsymbol{\pi}}(s_t) = \mathbb{E}_{\substack{\boldsymbol{a}_{t'} \sim \boldsymbol{\pi} \\ s_{t'} \sim \rho^{\boldsymbol{\pi}}}} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} R(s_{t'}, \boldsymbol{a_{t'}}) \right] \tag{1}$$

where $\rho^{\boldsymbol{\pi}} = \rho(s_0)\mathbb{E}_{\boldsymbol{a_t} \sim \boldsymbol{\pi}} \prod_{t=0}^{\infty} P(s_{t+1}|s_t, \boldsymbol{a_t})$ is the state distribution under joint policy $\boldsymbol{\pi}$. Additionally, the action-value function is defined to represent the value of the state-action pair

$$Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a_t}) = R(s_t, \boldsymbol{a_t}) + \mathbb{E}_{\substack{\boldsymbol{a}_{t'} \sim \boldsymbol{\pi} \\ s_{t'} \sim \rho^{\boldsymbol{\pi}}}} \left[\sum_{t'=t+1}^{\infty} \gamma^{t'-t} R(s_{t'}, \boldsymbol{a_{t'}}) \right]. \tag{2}$$

The advantage function is defined as $A^{\boldsymbol{\pi}}(s_t, \boldsymbol{a_t}) = Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a_t}) - V^{\boldsymbol{\pi}}(s_t)$. Let $\boldsymbol{c} = (c^i, \ldots, c_n)$ denote the joint experience, and the sequence $c^i = [(o_t^i, a_t^i, r_t^i, o_{t+1}^i)]_{t=0}^{T}$ of length T denote the local experience of agent i. Formally, the experience distribution is defined as follows:

$$p(\boldsymbol{c}|\boldsymbol{\pi}) = \rho(s_0) \prod_{t=0}^{T} P(s_{t+1}|s_t, \boldsymbol{a_t}) R(s_t, \boldsymbol{a_t}) \boldsymbol{\pi}(\boldsymbol{a_t}|\boldsymbol{o_t}). \tag{3}$$

It is worth noting that the above distribution depends directly on P and R of the environment. This connection enables us to infer the task at hand based on the observed experiences.

B. Single-Agent Context-Based Meta-RL

The purpose of context-based Meta RL is to derive a latent representation that encompasses task-specific features from the historical experience [10], [11]. Considering a task distribution $p(\mathcal{M})$, each task $M \sim p(\mathcal{M})$ is an MDP. Tasks share common state and action spaces, yet transition function P_M and reward function R_M may vary. The disparity between P_M and R_M is usually imperceptible directly. Agent must engage with the environment, inferring from sampled experiences. The task-specific experience is called *context* c_M as follows:

$$c_M \sim \rho(s_0) \prod_{t=0}^{T} P_M(s_{t+1}|s_t, a_t) R_M(s_t, a_t) \pi(a_t|o_t). \tag{4}$$

Agent learns a task encoder denoted as $q(z|c_M)$, which yields the task representation z conditioned on the context c_M.

In line with the previous works, the learning process is segmented into two phases: meta-training and meta-test [11], [12]. In the meta-training phase, the policy $\pi(a|o, z)$, critic $Q(s, a, z)$, and encoder $q(z|c_M)$ are concurrently trained. The meta-test phase is defined as executing N episodes on the same task, with the initial K episodes designated as exploration episodes and the subsequent $N - K$ episodes as evaluation episodes. During exploration, the policy network takes a prior representation $z \sim \mathcal{N}(\mathbf{0}, I)$ as input, aiming to collect task-specific context [29]. During evaluation, the task encoder will perform task inference based on the collected context, and the policy network takes the posterior $z \sim q(z|c_M)$ as input. The purpose of context-based meta-RL is to maximize the accumulated reward as follows:

$$\max \mathbb{E}_{\substack{M \sim p(\mathcal{M}) \\ a_t \sim \pi, s_t \sim \rho_M^{\pi}}} \left[\sum_{t=0}^{\infty} \gamma^t R_M(s_t, a_t) \right]. \tag{5}$$

IV. MIO-Based G2L Training Scheme

In this section, we present MG2L, a novel algorithm with an MIO-based G2L training scheme. As illustrated in Fig. 2, this algorithm extends the vanilla framework by introducing a multilevel task encoder. At the global level, the encoder aims to maximize the MI between the global representation and the task. At the local level, individual agents strive to learn a local representation that aligns with the global representation in terms of MIO.

In Section IV-A, we begin by formulating the context-based MAMRL, presenting the meta-CTDE framework and the structure of the multilevel task encoder. Following that, in Section IV-B, we propose the MG2L training scheme and the associated loss functions. Finally, in Section IV-C, we introduce a PIA module, and summarize the overall algorithm.

A. Multilevel Task Encoder in Meta-CTDE

Given the partial observability and nonstationarity of MASs, individual agent experiences often lack the depth required for adequate and efficient task representation extraction.

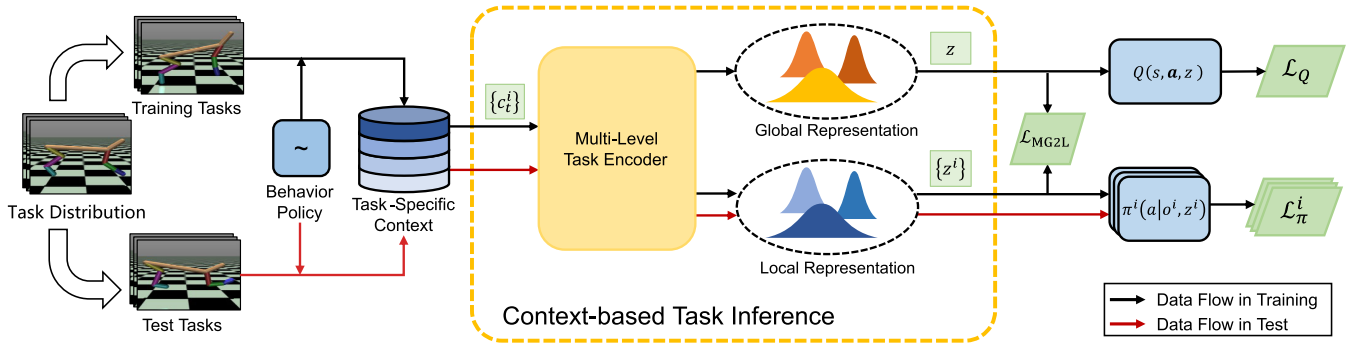

Fig. 2. Structure of MG2L. The left side illustrates the process of task sampling and context collection, where the task representation is sampled from the prior distribution $\mathcal{N}(\mathbf{0}, I)$ to ensure sufficient exploration. In the middle, the image demonstrates how the task representations z, z^i are extracted from the context. The multilevel task encoder performs global and local inferences, with the global inference results used for training the critic network and the local results for training the policy network. $\mathcal{L}_{\text{MG2L}}$ is the MG2L loss obtained through MIO, detailed in Section IV-B. The right side of the image shows how agents use the task representation for RL training and test.

To address this, we extend the CTDE framework to include the task inference component in MAMRL, forming the meta-CTDE framework as illustrated in Fig. 1(b). Similar to observation and action, the context can also be distinguished into local context $c^i = [o_t^i, a_t^i, r_t^i, o_{t+1}^i]_{t=0}^T$ and global one $\boldsymbol{c} = \{c^1, \dots c^n\}$. Then, the meta-CTDE can be formulated as follows:

$$\begin{aligned} &\text{CT}: Q(\boldsymbol{o}, \boldsymbol{a}, z), \quad z \sim q(z|\boldsymbol{c}) \\ &\text{DE}: \pi^i(a^i|o^i, z^i), \quad z^i \sim q^i(z^i|c^i). \end{aligned} \tag{6}$$

Each agent learns a decentralized local task encoder $q^i(z^i|c^i)$, and the conditional policy $\pi^i(a^i|o^i, z^i)$. While the agents share a centralized task encoder $q(z|\boldsymbol{c})$, and the critic $Q(\boldsymbol{o}, \boldsymbol{a}, z)$.

Following this formulation, we design the multilevel task encoder for context-based MAMRL. Given a global context $\boldsymbol{c} = \{[o_t^i, a_t^i, r_t^i, o_{t+1}^i]_{t=0}^T\}_{i=1}^n$, it can be viewed as a three-level hierarchy: transition level $c_t^i = (o_t^i, a_t^i, r_t^i, o_{t+1}^i)$, sequence level $c^i = \{c_t^i\}_{t=0}^T$, multiagent level $\boldsymbol{c} = \{c^i\}_{i=1}^n$. Correspondingly, the multilevel task encoder consists of the following components:

$$\begin{aligned} &\text{Transition Encoder: } x_t^i = E_{\text{tran}}(c_t^i) \\ &\text{Aggregation Encoder: } x^i = E_{\text{agg}}(x_0^i, \dots x_T^i) \\ &\text{Global Inference: } z \sim E_G(x^1, \dots x^n) \\ &\text{Local Inference: } z^i \sim E_L^i(x^i). \end{aligned} \tag{7}$$

After sampling contexts from the buffer, the transition encoder E_{tran} initially encodes each transition tuple into a single-agent single-timestep intermediate representation x_t^i. Subsequently, the aggregation encoder E_{agg} aggregates $\{x_t^i\}_{t=0}^T$ to single-agent multitimestep one x^i. Finally, on the inference, the global inference encoder E_G further aggregates $\{x^i\}_{i=1}^n$ to the global representation z, while the local encoder E_L^i encodes x^i to the local representation z^i. Let $E_G(E_{\text{agg}}(E_{\text{tran}}(\cdot)))$ be parameterized by ϕ_g, and E_L by ϕ_l^i. Then, the global inference is formulated as $z = E_{\phi_g}(c)$ and $z^i = E_{\phi_g, \phi_l^i}(c^i)$.

Although the sequence is denoted by the set $t = 0 : T$, it is important to note that the encoder operates based on transition tuples rather than trajectories. Furthermore, context sampling is random and decorrelated, aimed at reducing the impact of the behavioral policy on context so that the encoder can extract more robust task-specific features [11], [28]. In implementation, the transition encoder E_{tran} is realized by a multiple layer perceptron (MLP) and shared by all agents and all timesteps. However, there are various options for the aggregation encoder E_{agg} and global inference encoder E_G, such as the Gaussian Product [11] and sequence model [30], RNN [18]. In Section IV-C, we introduce a PIA module for the implementation of E_{agg} and E_G.

B. MIO-Based Global-to-Local Training Scheme

Building upon the introduced multilevel task encoder, we present the MIO-based G2L training scheme. We begin by designing the update scheme for global inference by maximizing the MI between global representation z and task M.

In context-based Meta-RL, there are two typical update schemes. The first, a straightforward scheme, relies solely on the RL losses [11], treating the task encoder as the front module of RL algorithm. Although it provides an end-to-end solution, RL update signals are stochastic and exhibit high variance, encompassing numerous task-redundant features that hinder efficient task inference. Hence, an additional auxiliary task is required. The second, as an auxiliary task, involves prediction losses [12], [18]. There is a prediction decoder $\hat{q}(\hat{s}_{t+1}, \hat{r}_t|z, s_t, a_t)$, and the loss function is based on the deviation between $(\hat{s}_{t+1}, \hat{r}_t)$ and (s_{t+1}, r_t). However, in a distributed MAS, due to the influence of decisions made by other agents, individual agents faces a nonstationarity environment. In such nonstationarity scenarios, prediction losses become unreliable because even with the same (o_t^i, a_t^i), different (o_{t+1}^i, r_t) may occur.

An ideal task representation should be able to extract sufficient task-specific features, remain independent of behavior policy, and filter out task-redundant features [26], [27], [28]. To achieve this objective, we introduce MIO. MI is a mathematical measure quantifying mutual dependence between two RVs. Computationally, it is equal to the reduction in uncertainty of one RV when the other RV is observed. In line with the concept of MIO, the task encoder is designed to maximize following objective:

$$I(z; M) = \mathbb{E}_{\mathcal{M}, z}\left[\log \frac{p(z|M)}{p(z)}\right] \tag{8}$$

which is the MI between the global representation z and the task M. The process of maximizing (8) aims to extract a comprehensive set of task-specific features, and simultaneously filters out task-redundant features.

Below, we provide a theorem that establishes a lower bound for $I(z; M)$ to design the update scheme. Here, for a global context query c sampled from task M, we define the context c^- sampled from different tasks $\mathcal{M}^- = \mathcal{M}\backslash M$ as negative samples. Inspired by noise contrastive estimation (InfoNCE) [14], we have the following theorem.

Theorem 1: Let $h(c; z) \triangleq {}^{p(z|c)}/_{p(z)}$ denote the density ratio, the MI between latent representation $z = E_{\phi_g}(c)$ and task M can be lower bounded by

$$I(z; M) \geq \mathbb{E}_{\mathcal{M},z,c}\left[\log \frac{h(c; z)}{\sum_{\mathcal{M}} h(c^*; z)}\right] + \log(N) \tag{9}$$

where N is the number of training tasks and c^* is the all positive and negative contexts.

Proof: We first convert the MI between latent representation z and task M into the one between z and context c

$$\begin{aligned} I(z; M) &= \mathbb{E}_{\mathcal{M},z}\left[\log \frac{p(z|M)}{p(z)}\right] \\ &= \mathbb{E}_{\mathcal{M},z}\left[\log \int_c \frac{p(z|c)p(c|M)}{p(z)} dc\right] \\ &= \mathbb{E}_{\mathcal{M},z}\left[\log \mathbb{E}_c \frac{p(z|c)}{p(z)}\right] \end{aligned} \tag{10}$$

where $p(z|c)$ is approximated by the multilevel task encoder $E_\theta(c)$. Since $\log(\cdot)$ is a concave function, using Jensen's inequality

$$I(z; M) - \log(N) \geq \mathbb{E}_{\mathcal{M},z,c}\left[\log \frac{p(z|c)}{p(z)}\right] - \log(N). \tag{11}$$

Split the denominator of the right hand side (RHS)

$$\begin{aligned} \text{RHS} &= \mathbb{E}_{\mathcal{M},z,c} \log\left[\frac{p(z|c)}{p(z) + (N-1)p(z)}\right] \\ &\geq -\mathbb{E}_{\mathcal{M},z,c} \log\left[1 + \frac{p(z)}{p(z|c)}(N-1)\right]. \end{aligned} \tag{12}$$

Note that the representation z is independent of the negative context c^-

$$\mathbb{E}_{\mathcal{M}^-}\left[\frac{p(z|c^-)}{p(z)}\right] = 1. \tag{13}$$

Estimate the expectation using the sum of samples

$$(N-1)\mathbb{E}_{\mathcal{M}^-}\left[\frac{p(z|c^-)}{p(z)}\right] \approx \sum_{\mathcal{M}^-} \frac{p(z|c^-)}{p(z)}. \tag{14}$$

Substitute (13) and (14) into (12)

$$\begin{aligned} \text{RHS} &\geq -\mathbb{E}_{\mathcal{M},z,c} \log\left[1 + \frac{p(z)}{p(z|c)} \sum_{\mathcal{M}^-} \frac{p(z|c^-)}{p(z)}\right] \\ &= \mathbb{E}_{\mathcal{M},z,c} \log\left[\frac{{}^{p(z|c)}/_{p(z)}}{{}^{p(z|c)}/_{p(z)} + \sum_{\mathcal{M}^-} {}^{p(z|c^-)}/_{p(z)}}\right] \\ &= \mathbb{E}_{\mathcal{M},z,c} \log\left[\frac{h(c; z)}{\sum_{\mathcal{M}} h(c^*; z)}\right] \end{aligned} \tag{15}$$

where $c^* = c \cup c^-$ denotes the all positive and negative contexts. That completes the proof. ■

According to (4), the distribution of the context $p(c|M)$ in (10) is also dependent on behavior policy $\boldsymbol{\pi}$. Therefore, when executing the transformation in (10), the behavior policy must remain fixed. This necessity is the reason why the policy and encoder buffer need to be updated simultaneously during meta-training. Furthermore, (11) implies that $I(z; c) \leq I(z; M)$, indicating that the process of collecting context may lead to information loss. The sufficiency of task-specific information implied in the context also depends on the exploration policy [31]. In implementation, exploration policy typically takes a prior representation as input to obtain more informative contexts [29]. In this article, we adopt this setting. Therefore, in the following, we will skip task M and directly delve into the MI between z and c.

Theorem 1 provides a lower bound for the MI between the global representation z and the task M. Note that $h(\cdot)$ cannot be calculated directly as a density ratio. In practice, it is common to use a similarity score function $g[\cdot]$ between query c and positive sample c^+ to estimate it, $h(c; z) \approx \exp[g[E_{\phi_g}(c^+); E_{\phi_g}(c)]]$.

Using the maximization of global MI as an auxiliary task, the loss function of global inference can be defined as follows:

$$\mathcal{L}_G(\phi_g) = -\frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}}\left[\log \frac{h(c; z)}{\sum_{\mathcal{M}} h(c^*; z)}\right] + \alpha D_{\text{KL}} \tag{16}$$

where $D_{\text{KL}} = D_{\text{KL}}[q(z|c)\|p(z)]$ is the Kullback-Leibler (KL) divergence regularization term, the prior distribution $p(z)$ is set to $\mathcal{N}(\mathbf{0}, I)$, and α is a hyperparameter. From a computational perspective, the first term in (16) is analogous to the cross-entropy loss commonly employed in multiclassification problems. If the query is correctly classified, indicating that the representation z is most similar to the positive one z^+, no loss is incurred. However, a loss is registered when the query is misclassified, i.e., when it is most similar to the representation z^- of a certain negative context. Therefore, this loss helps the encoder obtain more task-specific features and filter out task-redundant features.

Then, we formalize the RL loss of the global inference, and let critic Q be parameterized by θ_c, then the critic loss is defined as follows:

$$\mathcal{L}_Q(\phi_g, \theta_c) = \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}}\left[\hat{R}_t - Q(s_t, \boldsymbol{a}_t, z)\right]^2 \tag{17}$$

where $\hat{R}_t$ is the discounted reward-to-go.

The above loss function can effectively serve as the training objective for the critic and global inference in centralized meta-training. However, in MASs, agents cannot perform centralized inference due to partial observability. Therefore, we still need to provide training signals for local task inference. For local inference, a potential alternative is to integrate actor loss signals, but it is crucial to note that actor loss also demonstrates stochasticity and high variance. A better solution is to optimize the local inference by maximizing the MI between local inference and context, similar to global inference. While this scheme can still be further improved, it does not fully exploit the richer task-specific features encompassed

in the global representation. In light of these considerations, we introduce conditional MI $I(z^i; c|z)$ as the optimization objective to maximize the utilization of global representation. Due to the unknown nature of $p(c|z)$, we perform the following transformation. Take the definition of MI and conditional entropy, one has

$$\begin{aligned} I(z^i; c|z) &= H(z^i|z) - H(z^i|c, z) \\ &= -\big[I(z; c) - I(z, z^i; c)\big]. \end{aligned} \tag{18}$$

We define the conditional MI reduction as $I_r(z; z^i|c) = I(z; c) - I(z, z^i; c)$. Our key insight is to minimize $I_r(z; z^i|c)$ with the goal of narrowing the gap from G2L, thereby enhancing the quality of local representations. We accomplish this goal by optimizing the upper bound provided by the following theorem.

Theorem 2: The conditional MI reduction can be upper bounded by

$$I_r(z; z^i|c) \le I_{\mathrm{L1o}}(z; c) - I_{\mathrm{NCE}}(z, z^i; c) \tag{19}$$

where $I_{\mathrm{L1o}}(z; c) = \mathbb{E}_{z,c}[\log(h(c; z))/(\sum_{M^-} h(c^-; z))] + \log(N)$ and $I_{\mathrm{NCE}}(z, z^i; c) = \mathbb{E}_{z^i,c}[\log(h(c; z, z^i))/(\sum_{M} h(c^*; z, z^i))] + \log(N)$.

Proof: As the conditional MI reduction can be viewed as the difference between two items, we use the *leave one out bound* [23] to estimate the upper bound of $I(z; c)$, and InfoNCE [14] to estimate the lower bound of $I(z, z^i; c)$

$$\begin{aligned} I_{\mathrm{L1o}}(z; c) &= \mathbb{E}_{z,c} \log\left[N \frac{p(z|c)/p(z)}{\sum_{\mathcal{M}^-} p(z|c^-)/p(z)}\right] \\ &= -\mathbb{E}_{z,c} \log\left[\frac{1}{N} \frac{p(z)}{p(z|c)} \sum_{\mathcal{M}^-} \frac{p(z|c^-)}{p(z)}\right]. \end{aligned} \tag{20}$$

Similar to (14), substitute the sum with the expectation

$$I_{\mathrm{L1o}}(z; c) \approx -\mathbb{E}_{z,c} \log\left[\frac{N-1}{N} \frac{p(z)}{p(z|c)} \mathbb{E}_{\mathcal{M}^-} \frac{p(z|c^-)}{p(z)}\right]. \tag{21}$$

Combine (13) and (21)

$$\begin{aligned} I_{\mathrm{L1o}}(z; c) &\approx \mathbb{E}_{z,c} \log\left[\frac{N}{N-1} \frac{p(z|c)}{p(z)}\right] \\ &\ge \mathbb{E}_{z,c} \log\left[\frac{p(z|c)}{p(z)}\right] = I(z; c). \end{aligned} \tag{22}$$

The inequality $I(z, z^i; c) \le I_{\mathrm{NCE}}(z, z^i; c)$ can be easily proven by referring to Theorem 1. Then, it completes the proof. ■

Theorem 2 gives the upper bound of the conditional MI reduction. In practice, if the same batch of samples is used to estimate the expectation, (19) can be further simplified as follows:

$$L_{\mathrm{upper}} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \left\{ \log \frac{h(c; z)}{h(c; z, z^i)} + \log \frac{\sum_{\mathcal{M}} h(c^*; z, z^i)}{\sum_{\mathcal{M}^-} h(c^-; z)} \right\}. \tag{23}$$

For the first term, it represents the decrease in density ratio. For the second term, the numerator is the sum of the similarities of the negative pairs, whose ideal value is 0. The denominator is the sum of the similarities of all pairs, which is a regularization term.

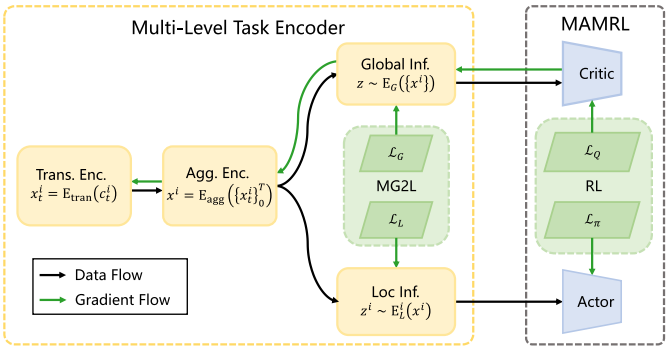

Fig. 3. Flow of data and gradients in the MG2L scheme. The losses consist of two parts: RL losses and MG2L losses. RL losses are used to update actors and critic, and $\mathcal{L}_Q(\theta_c, \phi_g)$ is backpropagated through every layer of the encoder. For MG2L losses, at the global level, $\mathcal{L}_G(\phi_g)$ is obtained by maximizing the MI between global representation and task, which is backpropagated to whole encoder. At the local level, $\mathcal{L}_L(\phi_l^i)$ is obtained by minimizing the conditional MI reduction, and it only acts on local inference.

Although the transition encoder and aggregation encoder are shared by global inference and local inference, they only use the global loss when updating. This ensures that the lower level encoders can also learn more global features. The loss function of local inference can be defined as follows:

$$\begin{aligned} \mathcal{L}_L(\phi_g, \phi_l^i) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \Bigg\{ &\log \frac{h(c; z)}{h(c; z, z^i)} \\ &+ \log \frac{\sum_{\mathcal{M}} h(c^*; z, z^i)}{\sum_{\mathcal{M}^-} h(c^-; z)} \Bigg\} + \alpha D_{\mathrm{KL}}. \end{aligned} \tag{24}$$

When estimating the density ratio and calculating the KL regularization, z and z^i are concatenated together to represent the joint distribution. Let actor π^i be parameterized by θ_a^i, then the actor loss is defined as follows:

$$\mathcal{L}_\pi(\theta_a^i) = -\frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \big[\mu_t^i A_t^i + \epsilon S[\pi^i(o_t^i, \mathrm{sg}(z^i))]\big] \tag{25}$$

where $A_t^i = \sum_{l=0}^{\infty} (\gamma\lambda)^l [r_{t+l} + \gamma V(s_{t+l+1}) - V(s_{t+l})]$ is computed by generalized advantage estimation (GAE) [32], $\mu_t^i = \pi_{\theta_a^i}(a_t^i|o_t^i, \mathrm{sg}(z^i)) / \pi_{\theta_a^i}(a_t^i|o_t^i, \mathrm{sg}(z^i))$, $S[\cdot]$ is the policy entropy regularization term, and ϵ are hyperparameters. Note that the policy loss is excluded from updating the encoder, due to its known biases and instability [11], [33].

The architecture of the flow of data and gradients in MG2L training scheme are illustrated in Fig. 3. Following the MG2L training scheme, the global inference learns how to extract the most task-specific features in the global context, and the local inference learns how to achieve a similar local representation in the absence of global experience.

C. Permutation-Invariant Attention Module

In Section IV-A, we introduced the multilevel task encoder, where the implementation of the aggregation encoder and the global inference encoder are optional. In the following, we introduce a PIA module for E_{agg} and E_G.

As shown in Fig. 4, E_{agg} and E_G are composed of N multi-head attention blocks and a PI head. The PIA module achieves permutation-invariance by eliminating position encoding, and

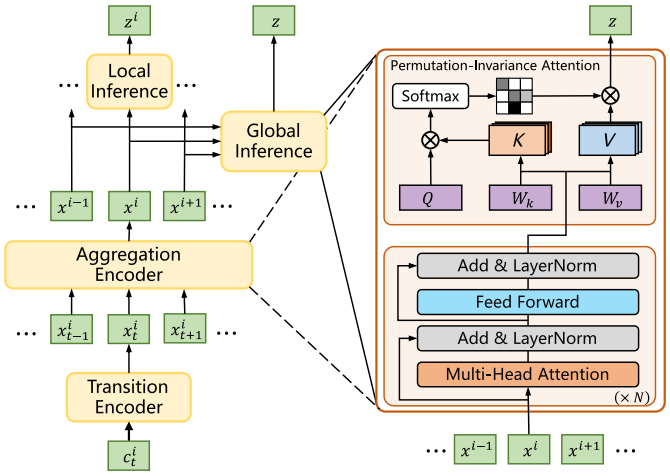

Fig. 4. Structure of multilevel task encoder and PIA module. Here, we employ the PIA module for E_{agg} and E_G.

it shares the query among all inputs in the PI head. Taking E_G as example, the output z via PIA module can be calculated by

$$z = E_G\left(\{x^i\}_{i=1}^n\right) = \sum_{i=1}^{n} \sigma\left(\frac{Q(\hat{x}^i W_k)^T}{\sqrt{d}}\right)\hat{x}^i W_v \tag{26}$$

where d is the dimension of latent space, $\sigma(\cdot)$ denotes the softmax operator, $\hat{x}^i$ is the output of multihead attention blocks, and $Q \in \mathbb{R}^{1\times d}, W_k, W_v \in \mathbb{R}^{d\times d}$ is the learnable parameters. Following the definition in [34], we have the following theorem.

Theorem 3: Let f_E denote the encoder mapping of PIA module, $\mathscr{S}$ denote the set of all permutations of $\mathcal{I} = \{1,\ldots,n\}$, $\forall \mathcal{P} \in \mathscr{S}$, $f_E(\mathcal{P}X) = f_E(X)$, where $X \in \mathbb{R}^{n\times d}$ denotes the vectorized inputs.

Proof: Without loss of generality, we assume an arbitrary permutation set $\mathcal{P} = \{p_1,\ldots,p_n\}$. H_P denotes the unique corresponding permutation matrix and e_{p_i} denotes the ith row of H_P, which is a one-hot row vector with the p_ith element equals to 1 and the rest equal 0. In that case, permuting the rows of a matrix is equivalent to left-multiplying it by a matrix H_P, let $K_P = H_P X W_k \in \mathbb{R}^{n\times d}$, $V_P = H_P X W_v$ denote the permuted keys and values. Besides, $\forall i, \|e_i\| = 1$ and $e_i^T e_j = 0$ when $i \neq j$, which indicates H_P is an orthogonal matrix, $H_P^T H_P = I$.

The output Z_P can be calculated by

$$Z_P = \sigma\left(QK_P^T\right)V_P = \sigma\left(QK^T H_P^T\right)H_P V. \tag{27}$$

Note that $\sigma(\cdot)$ operator is a vectorized operator and cannot be directly interchanged with H_P. So, we first prove the following equation:

$$\sigma\left(QK^T H_P^T\right) = \sigma\left(QK^T\right)H_P^T. \tag{28}$$

Expand the matrix equation of left hand side (LHS), substitute $\sigma(\cdot)$ operator with its definition

$$\begin{aligned}\text{LHS} &= \sigma\left(\left[\begin{array}{ccc} QK^T e_{p_1}^T & \cdots & QK^T e_{p_n}^T \end{array}\right]\right)\\ &= d_L\left[\begin{array}{ccc} \exp\left(QK[p_1]^T\right) & \cdots & \exp\left(QK[p_n]^T\right)\end{array}\right]\end{aligned} \tag{29}$$

where $d_L = 1/\sum_{i=1}^n \exp(QK^T e_{p_i}^T)$, and $K[i] \in \mathbb{R}^{1\times d}$ denotes the ith row of K. Perform the same expansion on the RHS

$$\begin{aligned}\text{RHS} &= d_R\left[\begin{array}{ccc} \exp(QK[1]^T) & \cdots & \exp(QK[n]^T)\end{array}\right]H_P^T\\ &= d_R\left[\begin{array}{ccc} \exp\left(QK[p_1]^T\right) & \cdots & \exp\left(QK[p_n]^T\right)\end{array}\right]\\ &= \text{LHS}\end{aligned} \tag{30}$$

where $d_R = \sum_{i=1}^n \exp(QK[i]^T) = \sum_{i=1}^n \exp(QKe_{p_i}^T) = d_L$. Substitute (28) into (27)

$$Z_P = \sigma\left(QK^T\right)E_P^T E_P V = \sigma\left(QK^T\right)V = Z. \tag{31}$$

This completes the proof. ■

Theorem 3 ensures that the encoder is permutation-invariant at both aggregation and global inference levels. As mentioned before, the distribution of task-specific features in the context is uneven, with the experiences of certain agents or at certain timesteps playing a more crucial role in task inference. At the aggregation level, the encoder pays more attention to the transitions that encompass more task-specific features, enhancing the efficiency of the agents, especially in scenarios with sparse reward signals. Additionally, the introduction of the PIA module allows the encoder to take decorrelated transition tuples as input, instead of trajectories, diminishing the impact of the behavior policy and enhancing the robustness of task representation. At the global inference level, the encoder utilizes the local experiences of all different agents, thereby addressing the challenge of partial observability. The global representation contributes to a more comprehensive understanding by considering the diverse perspectives of individual agents. This comprehensive understanding is then propagated to local inference through the MG2L training scheme described in Section IV-B.

Furthermore, we introduce a priority buffer based on the PIA module. Note that (26) is a weighted average based on attention scores. These attention scores reflect the contribution of transitions to task inference. We take the attention score ω_t^i as transition priority

$$\omega_t^i = \sigma\left(\frac{Q^{\text{agg}}\left(\hat{x}_t^i W_k^{\text{agg}}\right)^T}{\sqrt{d}}\right) \tag{32}$$

where $Q^{\text{agg}}, W_k^{\text{agg}}$ is the parameters of aggregation encoder. In the meta-test phase, the encoder initially assigns scores ω_t^i to the collected transitions. During evaluation, transitions with higher attention scores have a greater probability of being selected. This allows agents to adapt to new tasks more quickly and effectively. This mechanism facilitates agents adapting to new tasks more rapidly and effectively.

The brief procedure of centralized meta-training and decentralized meta-test are summarized in Algorithms 1 and 2, respectively.

V. Experiments

In this section, we begin by introducing the environmental and task settings employed in the experiments. Subsequently, we conduct comparative evaluations, where we assess the performance of our algorithm MG2L against other baseline methods. Then, we delve into an analysis of their generalization process in meta-test, and provide visualizations of the learned task representations. In terms of ablation studies, we explore the impact of the losses and the training scheme in MG2L, thereby confirming the effectiveness of the proposed

Algorithm 1 Centralized Meta Training

Input: Batch of training tasks $\{M_k\}_{k=1...T}$ sampled from distribution $p(\mathcal{M})$, context buffer $\mathcal{C}_k$, RL buffer $\mathcal{B}_k$ for each task M_k, parameterized networks, $\theta_a^i, \theta_c, \phi_g, \phi_l^i$ for policy π^i, critic Q and multilevel task encoder

while not done **do**
  # Collect transitions for context
  for each task M_k, timestep t **do**
    Sample prior task representation $z_t^i \sim \mathcal{N}(\mathbf{0}, I)$
    Rollout policy $\prod_i \pi^i(a_t^i|o_t^i, z_t^i)$ on task M_k
    Store transition tuple into $\mathcal{C}^k$
  end for
  # Collect transitions for RL
  for each task M_k, timestep t **do**
    Sample context c^i from $\mathcal{C}^k$, $z_t^i = E_\psi^i(c^i)$
    Rollout policy $\prod_i \pi^i(a_t^i|o_t^i, z_t^i)$ on task M_k
    Store transition tuple into $\mathcal{B}^k$
  end for
  # Update networks
  for each task M_k **do**
    Sample RL batch from $\mathcal{B}^k$
    Update θ_c, ϕ_g to minimize $\mathcal{L}_Q(\theta_c, \phi_g) + \mathcal{L}_G(\phi_g)$
    Update θ_a^i to minimize $\sum_i \mathcal{L}(\theta_a^i)$
    Update ϕ_l^i to minimize $\sum_i \mathcal{L}_L(\phi_l^i)$
  end for
end while

Algorithm 2 Decentralized Meta Test

Input: Trained models $E_{\phi_g,\phi_l^i}, \pi^i$, test task M, priority buffer $\mathcal{C}$

 # Exploration
for episode$= 1 : K$, timestep t **do**
  Sample prior task representation $z_t^i \sim \mathcal{N}(\mathbf{0}, I)$
  Rollout policy $\prod_i \pi^i(a_t^i|o_t^i, z_t^i)$ on task M
  Store transition tuple into $\mathcal{C}$
  Update transition priority ω_t^i
end for
 # Evaluation
for episode$= K+1 : N$, timestep t **do**
  Sample context c^i based on ω_t^i from $\mathcal{C}$, $z_t^i = E_\psi^i(c^i)$
  Rollout policy $\prod_i \pi^i(a_t^i|o_t^i, z_t^i)$ on task M
  Get reward r_t and next observation o_{t+1}^i
end for

approach. Finally, we conduct ablation studies on the network structure to further validate the effectiveness of the PIA module. For more details on the implement of MG2L, please refer to https://github.com/zhaozijie2022/mg2l.

A. Experimental Settings

As demonstrated in Fig. 5, we evaluate the proposed method MG2L on a variety of continuous and discrete control tasks. Their descriptions are as follows.

1) *MA-HalfCheetah-Dir:* HalfCheetah is a continuous control task on the Mujoco simulator, where the agent controls the torque of each joint with the goal of making the cheetah robot run as fast as possible. HalfCheetah-Dir is a widely used multitask variant [9], [10], [11], [12], [26], [27], [28], where the task is specified by the goal direction, which can be forward or backward. Here, we employ multiagent Mujoco Benchmark [35], a widely used multiagent extension of Mujoco.
2) *Spread-Target:* Spread is a cooperative task within the multiagent particle environment (MPE) [36], where agents receive rewards based on their distances from various landmarks. In Spread-Target, the number of landmarks m is greater than the number of agents n; however, only n landmarks are effective in each task, meaning they are the ones that generate rewards. Agents can observe the positions of all landmarks but need to contextual infer which landmarks are effective.
3) *Hunting-Target:* Hunting (Predator-Prey) is a competitive task within MPE, where the goal of the n predators is to capture a single prey. Similar to Spread-Target, multiple preys exist, but only one is effective in each task. This requires agents to contextually infer which prey is the designated target.
4) *Rware-Layout:* Rware (multirobot warehouse) is a cooperative discrete control task [37], where agents are required to pick up requested items from shelves and deliver them to the gate. The task is specified by the layout of the shelves and the quantity of orders [18].
5) *MA-Hopper-Param:* Hopper is a continuous control task on Mujoco, for which we employ MA-Mujoco. The task of Hopper-Param [9], [10], [11], [12], [26], [27], [28] is specific by the physical parameters of mass, inertia, damping, and friction. The agents should adapt to the varying physical parameters and learn to hop as fast as possible.
6) *MAgent-Gather:* MAgent [38] is a large-scale MARL platform, where gather is one of the most widely used environments. In gather, controllable agents (blue units) are rewarded for eliminating all foods (blue units) as quickly as possible. The task is specified by the layout and density of foods.

For each experiment, the estimated per-episode performance is averaged over at least five random seeds. The detailed hyperparameters and computational costs are provided in Appendix A. For more detailed information on the multitask setting, refer to Appendix B.

B. Comparative Evaluation

We report the average return per episode of the proposed algorithm MG2L in the above six environments, compared with mixture-MATE (Mix-MATE) [18], [20], multiagent-PEARL (MA-PEARL) [11], and task-given method. MATE is notable as the first algorithm that introduces task representation in MAMRL. MATE provides three versions: cen centralized training and centralized execution (CTCE), ind decentralized training and decentralized execution (DTDE), and mix (CTDE), the mix-MATE is employed to align with the CTDE in MG2L. MA-PEARL is a direct multiagent extension of PEARL, where the local representations z^i remain consistent with the single-agent version, while the critic utilizes a

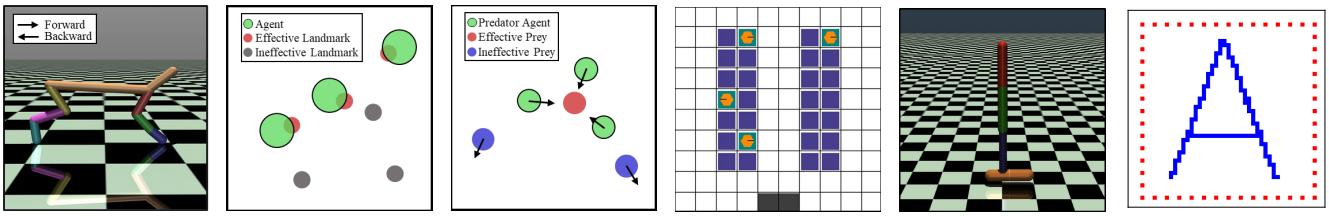

Fig. 5. Demonstrations of the meta multiagent environments. (a) HalfCheetah-Dir. (b) Spread-Target. (c) Hunting-Target. (d) Rware-Layout. (e) Hopper-Param. (f) MAgent-Gather.

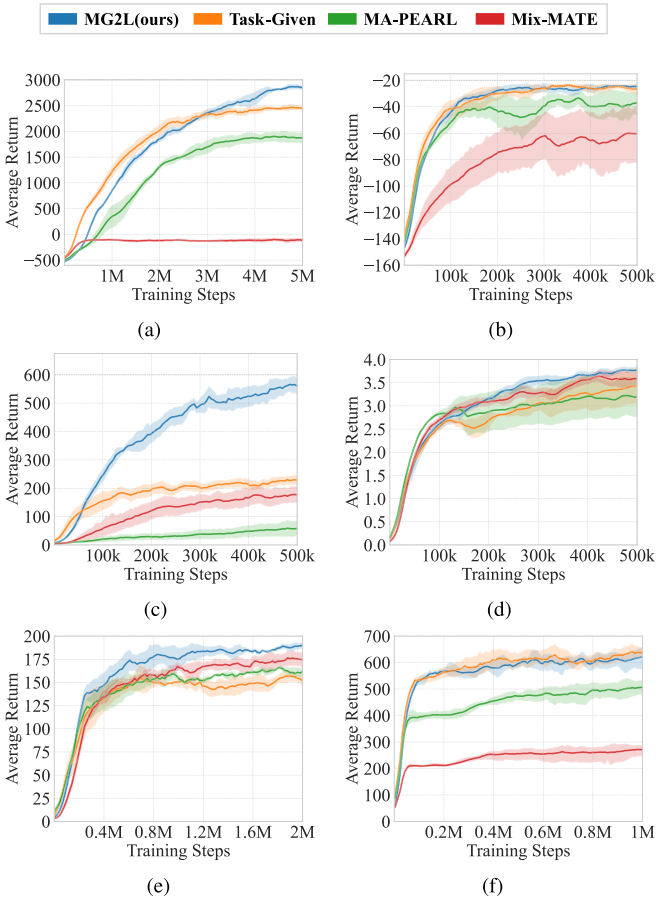

Fig. 6. Comparisons against the baselines on the various environments. Here, the tasks of environments in Fig. 5(a)–(c) are specific by reward functions, and the tasks of environments in Fig. 5(d)–(f) are specific by transition dynamics. Solid curves represent the mean performance over five random seeds, and shaded regions represent the confidence intervals. (a) HalfCheetah-Dir. (b) Spread-Target. (c) Hunting-Target. (d) Rware-Layout. (e) Hopper-Param. (f) MAgent-Gather.

Gaussian product of $\{z^i\}_{i=1}^n$ as global representation. The task-given method represents an ideal setting where the agent can directly access task labels without the need for inference. Note that the task labels as ground truth are generally unavailable. The codes of these baselines are all open-source. For fairness, we replace the MARL algorithms of baselines and MG2L with multi-agent proximal policy optimization (MAPPO) [39]. The hyperparameters are fixed across different methods.

Fig. 6 presents the results of the comparative evaluation, where MG2L consistently outperforms other methods in almost all environments In environments in Fig. 5(a)–(c), where tasks are defined by reward functions necessitating the extraction of distinctive features, MG2L particularly excels. In the challenging Hunting-Target environment, MG2L demonstrates significantly superior performance, attributed to its effective utilization of global context, enabling the MAS to better address the challenge of partial observability. In Halfcheetah-Dir and Hunting-Target, MG2L exhibits a slightly slower initial rise compared to the task-given method, indicating the learning process of the encoder to distinguish between different tasks. However, in later stages, MG2L surpasses even the task-given method. This is due to the encoder's ability to extract richer features beyond task labels. In environments in Fig. 5(d)–(f), the tasks are specified by transition dynamics, and the encoder is required to possess adaptability and generalization to varying parameters. The suboptimal performance of the task-given method is attributed to the limitations of task labels, which only identify tasks without generalization. The excellent performance of MG2L on MAgent-gather shows that the algorithm exhibits strong scalability for the learning tasks with a large number of agents. This is attributed to the design of the multilevel task encoder and the introduction of the PIA module, which reduce the computational complexity associated with processing large-scale tasks and enable the agent to handle global context more efficiently.

Additionally, the shaded regions indicate that MG2L achieves superior performance while maintaining higher training stability compared to the baselines.

C. Meta Test Evaluation

We evaluate the generalization performance of MG2L and the baselines during meta-test, and the results are reported in Fig. 7. MG2L demonstrates significantly faster generalization across all environments, which is mainly attributed to the efficient utilization of local context. Particularly, in the Hunting-Target environment, the agent's observations are limited by the observation radius. When the target is not present in the field of view, the accumulated context mainly consists of task-redundant features. This scenario puts a stronger emphasis on the algorithm's ability to focus on task-specific features and effectively filter out task-redundant features. Fig. 7(c) shows that both MA-PEARL and Mix-MATE show almost no increase within the first 100 steps of exploration, while MG2L's rapid rise reaffirms its ability to quickly focus and extract task-specific features from a large amount of redundant information. Fig. 7(d) shows the accuracy of task inference during meta-test, which follows the same trend as the average return.

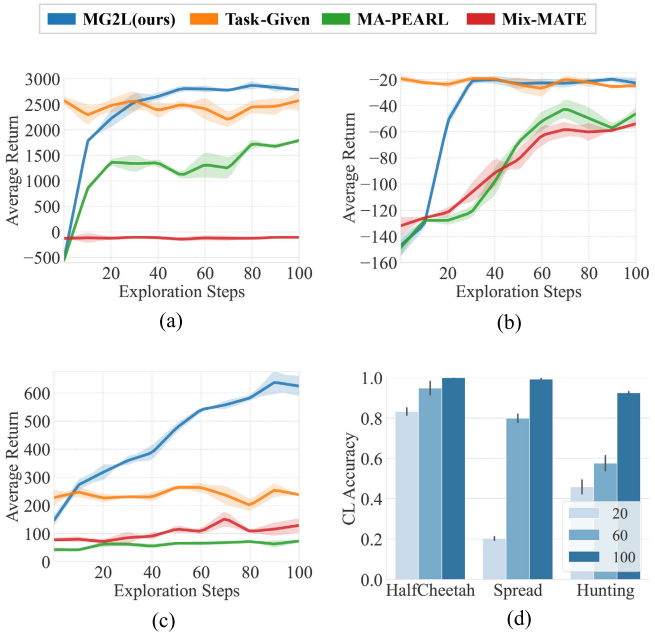

Fig. 7. Generalization performance in meta-test. (a) HalfCheetah-Dir. (b) Spread-Target. (c) Hunting-Target. (d) CL accuracy.

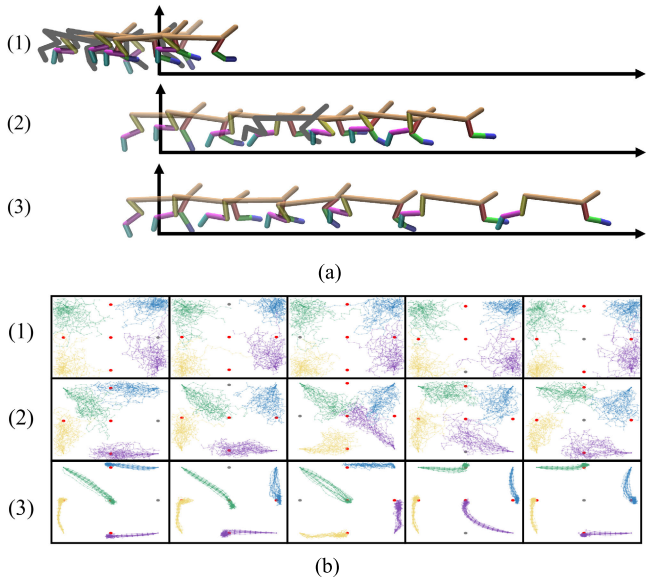

Fig. 8. Visualization of the meta-test process of MG2L. Here, (1) corresponds to the result without exploration, i.e., z^i follows the prior distribution; (2) corresponds to the performance during exploration; and (3) corresponds to the converged result after sufficient exploration. In HalfCheetah-Dir, the task is set to run forward, and the initial timestep is indicated by the ordinate in the figure. The frames with backward velocity are marked in gray, and those with forward velocity are not processed. In Spread-Target, for clarity, we fix the starting points of the agents and the positions of the landmarks, where the effective coordinates are marked in red and the ineffective ones are gray. (a) HalfCheetah-Dir. (b) Spread-Target.

Fig. 8 provides further visualization of the meta-test process. In HalfCheetah-Dir, as exploration progresses, the agents gradually learn to move forward and eventually converge to optimal strategies. Similarly, in the Spread-Target, the trajectories gradually converge to effective landmarks.

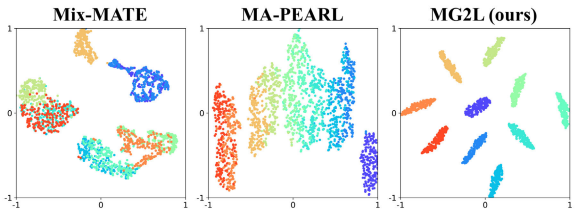

Fig. 9. 2-D visualization of the learned task representation space via t-distributed stochastic neighbor embedding (t-SNE). Each point corresponds to a task representation, where the same color indicates that the context comes from the same task.

D. Visualization of Task Representation

We sampled ten tasks from the task distribution of Hopper-Param, obtaining 100 sets of contexts for each task. The visualization of the learned task representations is presented in Fig. 9. In comparison to other methods, MG2L effectively distinguishes different tasks while aggregating representations from the same task. This indicates that the encoder can extract more discriminative features. These more robust representations will assist the agents in achieving better performance during meta-test.

E. Analysis on Context Mismatch

We conduct an analysis on context mismatch gap in Hopper-Param, to measure the differences between meta-tasks and emphasize the importance of accurate task representation. We meta-train MG2L on this task distribution. During meta-test, ten tasks are uniformly sampled, and 100 (source, target) task pairs are created. Agents collect context from the source tasks and then perform inference and generalization on the target tasks. Please note that this is not a zero-shot test, rather it resembles the "negative transfer" setting found in transfer learning [40], aimed at assessing how much the source task-specific context hinders the performance on target tasks. Let M_i and M_j denote the source and target tasks, the context mismatch performance is defined as

$$R[(j),(i)] = \mathbb{E}_{\substack{a\sim\pi(\cdot|o,z(i)) \\ s'\sim P(j)}}\left[\sum_{t=0}^{T}\gamma^t r_t(j)\right] \tag{33}$$

where $z(i) \sim q(\cdot|c(i))$ denotes the task representation of the source task M_i, $P(j)$ and $r_t(j)$ denote the transition dynamics and reward of the target task M_j. The results are reported in Fig. 10.

The distribution of meta-tasks in the parameter space shown in Fig. 10(a), which can serve as a reference for understanding the differences between meta-tasks. The matrices in Fig. 10(b)–(e) reflect a similar pattern with Fig. 10(a), with higher test performance along the main diagonal that gradually declines toward the sides. However, due to the nonlinearity and complexity of the dynamics transition function, two task pairs with similar distances in parameter space may still exhibit significant differences in meta-test performance.

First, variations in a single parameter do not translate linearly in meta-learning. For example, when the source task is with a small mass, the model struggles to generalize to other tasks. In contrast, when the source task is with a large

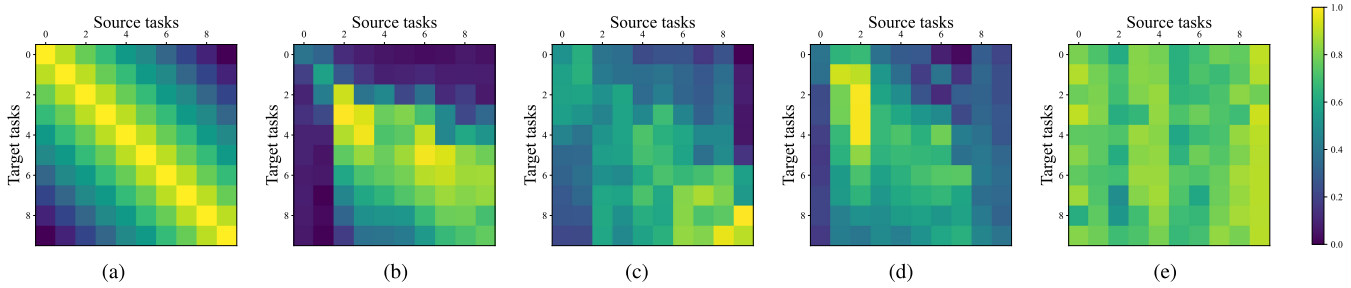

Fig. 10. Gap on Hopper-Param. Matrix in (a) shows the distribution of meta-tasks in the parameter space. Matrices in (b)–(e) show the context mismatch gap under different parameters. Here, the color represents the normalized average return. The main diagonal of the matrix signifies instances where the source and target tasks align; the farther from the main diagonal, the greater the difference between the meta-tasks in the parameter space. (a) Parameter space. (b) Mass. (c) Damping. (d) Friction. (e) Inertia.

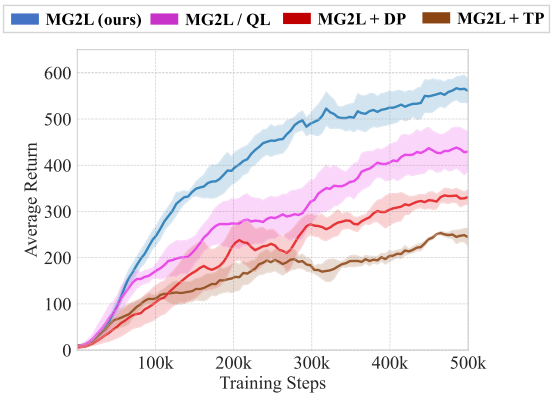

Fig. 11. Ablation results on the global loss.

mass, the model's generalization capability is stronger across a wider range of tasks. Moreover, different parameters affect meta-learning to different extents. The results in Fig. 10(e) suggest that inertia has a smaller effect on meta-learning compared to parameters like mass, damping, and friction.

F. Ablation of G2L Training Scheme

In this section, we evaluate the effectiveness of MG2L through ablation studies on the global loss, local loss, and training scheme.

We first maintain the local loss and training scheme unchanged, perform an ablation on the global loss. This involves evaluating the impact of the introduced CL in Section IV-B by comparing it with other global loss designs. Here, we set up four experimental groups. MG2L(ours) represents the original setting, where the global loss consists of Q loss and CL loss; MG2L/QL retains only Q loss, serving as a blank control group; MG2L + DP replaces the introduced CL loss with dynamics prediction loss; MG2L + TP replaces it with task label prediction loss, as a supervised learning control group.

Experimental results are reported in Fig. 11. The findings emphasize that inappropriate auxiliary losses may hinder the extraction of task-specific features. For example, the introduction of DP loss negatively impacts performance due to the presence of many task-redundant features in the dynamics. Similarly, the introduction of TP loss leads to a significant decline as one-hot-encoded task labels inherently lack task semantics. In contrast, CL loss facilitates the efficient extraction of task-specific features, resulting in optimal performance.

In the next step, we keep the global loss unchanged and perform ablations on the local loss and training scheme, and the results are reported in Fig. 12. Here, Global-Only represents an ideal setting under CTCE, where all agents have access to global context and directly use the global representation z as the input for the policies. Local-Only is a naive implementation under CTDE, where z is concatenated with all z^i. MG2L w CSTL is an alternative to G2L, where (24) is replaced with a **C**on**S**is**T**ency **L**oss, specifically the KL divergence between $q(z^i|c^i)$ and $q(z|c)$. The introduction of the consistency loss aims to make the two distributions as close as possible. The training schemes for different experimental groups are illustrated in Fig. 12(a).

The experimental results in Fig. 12(b) demonstrate that MG2L achieves performance comparable to Global-Only under CTCE, far exceeding the other two control groups. Fig. 12(c) explains the effectiveness of MG2L, indicating that the MI-based local loss in MG2L converges quickly after the global loss converges, whereas the MI loss in Local-Only does not converge. This suggests that relying solely on local context cannot construct sufficiently distinctive representations. Furthermore, Fig. 12(d) shows the consistency loss in MG2L w CSTL, with a rapid decline in the early stages but an inability to converge in the later stages. This indicates that the scarce and uneven task-specific information in the local context is insufficient to support local representations to fully replicate global representations. Therefore, it is reasonable and effective for MG2L to choose MI as the intermediate medium for distillation.

G. Ablation of PIA Module

As mentioned in Section IV-A, MG2L utilizes a multi-level task encoder, and the PIA module is introduced in Section IV-C. We conduct an ablation study on the aggregator's structure while retaining the MG2L training scheme unchanged. The PIA module is compared with several alternative models, including Gaussian Product [11], transformer [30], and RNN [10], and the results are reported in Fig. 13.

PIA module achieves optimal performance with the highest learning speed. In the control groups, the transformer achieves

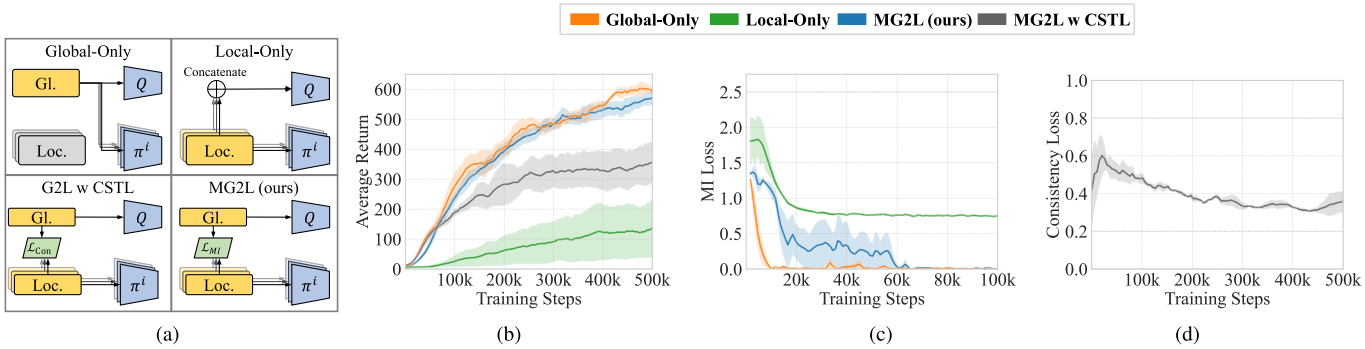

Fig. 12. Ablation results on the local loss and training scheme. (a) Scheme setting. (b) Average return. (c) Convergence of MI loss. (d) Convergence of consistency loss.

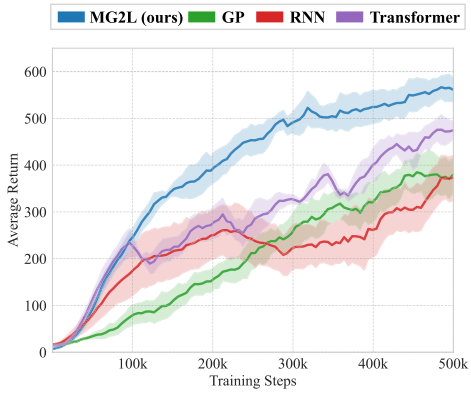

Fig. 13. Comparisons of different aggregators on the modified Hunting-Target environment, where GP denotes Gaussian product. For all experiments, we sample context from decorrelated transition tuples, which is more efficient and robust than sampling trajectories.

TABLE I
NETWORK CONFIGURATIONS

Task Encoder	Value	Actor & Critic	Value
hidden sizes	[64, 64]	hidden sizes	[128, 128]
num of blocks	2	gain	0.01
num of heads	4	feature norm	true
task representation dim	64	value norm	true
activation	ReLU	activation	ReLU

TABLE II
TRAINING HYPERPARAMETERS

Hyperparameters	Value	Hyperparameters	Value
training threads	32	rollout threads	32
share policy	false	learning rate	5e-4
optim eps	1e-5	optimizer	Adam
weight decay	0	ppo clip	0.2
num of mini batch	1	entropy coef ϵ	0.01
max grad norm	10	gae λ	0.95
reward discount γ	0.99	D_{KL} weight α	0.1
training epochs	15	lr decay	Linear

a learning speed comparable to the PIA module, but its lack of permutation-invariance leads to lower stability due to interference information in the sequence. Conversely, the Gaussian Product exhibits the opposite scenario, as it assigns equal importance to all inputs. This characteristic makes it unable to effectively filter task-redundant features, resulting in a lower learning speed. While for RNN, even when provided

TABLE III
ENVIRONMENT-SPECIFIC HYPERPARAMETERS AND COMPUTATIONAL COSTS

	(a)	(b)	(c)	(d)	(e)	(f)
training steps	5M	500K	500K	500K	2M	1M
buffer size	1M	100K	100K	100K	500K	200K
batch size	512	32	32	64	512	128
eval interval	500K	50K	50K	50K	200K	100K
eval episodes	32	32	32	32	32	32
training time	1.7h	1h	1h	1h	1.2h	5.2h

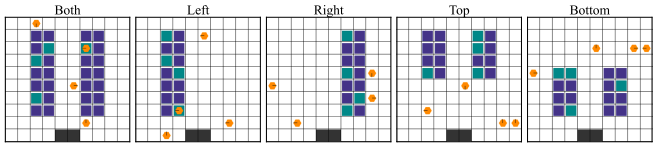

Fig. 14. Multitask layouts of shelves for multirobot warehouse environments. Here, the agent's position and direction are generated randomly.

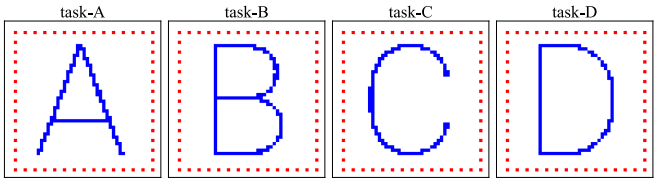

Fig. 15. Multitask layouts of foods for MAgent-Gather environments.

with decorrelated context as input, its performance remains inferior and unstable, which is attributed to the inability of RNN to extract task-specific features in complex scenarios.

VI. CONCLUSION

This article presents MG2L, a novel approach designed to address the task representation challenge in MAMRL. Leveraging a multilevel task encoder and the MIO-based G2L training scheme in meta-CTDE, MG2L efficiently extracts task-specific features. This results in optimal performance across various multitask environments, particularly exceling in challenging partially observable settings. Utilizing CL, MG2L rapidly focuses on task-specific features, thereby enhancing learning efficiency. The MIO-based G2L training scheme effectively addresses the scarcity and unevenness of task-specific features in local context, achieving efficient G2L distillation. Additionally, we propose a PIA module that

accelerates the process of focusing on and extracting task-specific features, reducing the impact of policy variations on task representation, thereby improving learning speed and stability.

APPENDIX A
HYPERPARAMETERS AND COMPUTATIONAL COSTS

The network configurations, training hyperparameters, and environment-specific hyperparameters are detailed in Tables I–III, respectively. Please note that approximate the MG2L computational costs are reported in hours on a single NVIDIA GeForce RTX A6000 GPU, as shown in Table III.

APPENDIX B
DETAILED MULTITASK SETTINGS

For environments where the task is specified by the reward function, we can directly provide the analytical reward function as the task definition.

1) For HalfCheetah-Dir

$$r_t = v_t \cdot g_M - 0.01 \cdot ||\boldsymbol{a}||^2$$

where v_t is the velocity of the cheetah robot, g_M is the task-specific parameter, which is set to $+1$ when the task is "forward" and -1 when it is "backward," and $||\boldsymbol{a}||^2$ is the action penalty.

2) For Spread-Target

$$r_t = -\sum_{j=1}^{m} g_M^j \min_i (p_i - p_j)^2$$

where p_i is the position of agent i, p_j is the position of landmark j, and g_M^j is the task-specific parameter for landmark j, which is set to $+1$ when the landmark is effective and 0 otherwise.

3) For Hunting-Target

$$r_t = -\sum_{j=1}^{m} g_M^j (p_i - p_j)^2$$

where g_M^j is the task-specific parameter for prey j, which is set to $+1$ when the prey is effective and 0 otherwise.

For environments where the task is specified by the dynamics transition function, since the dynamics transition function is often implicit and nonanalytical, we can define tasks by specifying key task-specific parameters.

4) For Rware-Layout, we visualize all the layouts in Fig. 14.
5) For Hopper-Param, its key task-specific parameters are the robot's physical parameters: mass, inertia, damping, and friction.
6) For MAgent-Gather, we visualize all the layouts in Fig. 15.

REFERENCES

[1] R. Lu, Y. Zhu, D. Zhao, Y. Liu, and Y. He, "Last-iterate convergence to approximate Nash equilibria in multiplayer imperfect information games," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2024.

[2] Z. Tang, Y. Zhu, D. Zhao, and S. M. Lucas, "Enhanced rolling horizon evolution algorithm with opponent model learning: Results for the fighting game AI competition," *IEEE Trans. Games*, vol. 15, no. 1, pp. 5–15, Mar. 2023.

[3] Y. Fu, Y. Zhu, J. Chai, and D. Zhao, "LDR: Learning discrete representation to improve noise robustness in multiagent tasks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 1, pp. 513–525, Jan. 2025.

[4] J. Chai, Y. Zhu, and D. Zhao, "NVIF: Neighboring variational information flow for cooperative large-scale multiagent reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17829–17841, Dec. 2024.

[5] L. Chen, B. Hu, Z.-H. Guan, L. Zhao, and X. Shen, "Multiagent meta-reinforcement learning for adaptive multipath routing optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5374–5386, Oct. 2022.

[6] Z. Zhao, L. Shi, T. Li, J. Shao, and Y. Cheng, "Opinion dynamics of social networks with intermittent-influence leaders," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 3, pp. 1073–1082, Jun. 2023.

[7] J. Chai et al., "UNMAS: Multiagent reinforcement learning for unshaped cooperative scenarios," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2093–2104, Apr. 2023.

[8] A. Mahajan et al., "Generalization in cooperative multi-agent systems," 2022, *arXiv:2202.00104*.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.

[10] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "RL^2: Fast reinforcement learning via slow reinforcement learning," 2016, *arXiv:1611.02779*.

[11] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Jun. 2019, pp. 5331–5340.

[12] Y. Mu et al., "DOMINO: Decomposed mutual information optimization for generalized context in meta-reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35. Red Hook, NY, USA: Curran Associates, 2022, pp. 27563–27575.

[13] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*, vol. 1. Cham, Switzerland: Springer, 2016.

[14] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[15] Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *J. Autom. Intell.*, vol. 3, no. 2, pp. 73–91, Jun. 2024.

[16] F. Zhang, C. Jia, Y.-C. Li, L. Yuan, Y. Yu, and Z. Zhang, "Discovering generalizable multi-agent coordination skills from multi-task offline data," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–12.

[17] W. Mao et al., "Multi-agent meta-reinforcement learning: Sharper convergence rates with task similarity," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–22.

[18] L. Schäfer, F. Christianos, A. Storkey, and S. V. Albrecht, "Learning task embeddings for teamwork adaptation in multi-agent reinforcement learning," 2022, *arXiv:2207.02249*.

[19] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2681–2690.

[20] L. Schäfer, "Task generalisation in multi-agent reinforcement learning," in *Proc. 21st Int. Conf. Auto. Agents Multiagent Syst.*, 2022, pp. 1863–1865.

[21] T. Hoang, T.-T. Do, T. V. Nguyen, and N.-M. Cheung, "Multimodal mutual information maximization: A novel approach for unsupervised deep cross-modal hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6289–6302, Sep. 2023.

[22] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2016, pp. 1–12.

[23] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Jun. 2019, pp. 5171–5180.

[24] N. Li, Y. Chen, W. Li, Z. Ding, D. Zhao, and S. Nie, "BViT: Broad attention-based vision transformer," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12772–12783, Sep. 2024.
[25] L. Ren, Z. Jia, Y. Laili, and D. Huang, "Deep learning for time-series prediction in IIoT: Progress, challenges, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15072–15091, Nov. 2024.
[26] L. Li, R. Yang, and D. Luo, "FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–20.
[27] H. Fu et al., "Towards effective context for meta-reinforcement learning: An approach based on contrastive learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 7457–7465.
[28] H. Yuan and Z. Lu, "Robust task representations for offline meta-reinforcement learning via contrastive learning," in *Proc. 39th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 132, Jan. 2022, pp. 25747–25759.
[29] Z. Wang, C. Zhang, and K. Chaudhuri, "Thompson sampling for robust transfer in multi-task bandits," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, Jan. 2022, pp. 23363–23416.
[30] L. C. Melo, "Transformers are meta-reinforcement learners," in *Proc. 39th Int. Conf. Mach. Learn.*, vol. 162, Jan. 2022, pp. 15340–15359.
[31] P. Jiang, S. Song, and G. Huang, "Exploration with task information for meta reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4033–4046, Aug. 2023.
[32] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.
[33] J. Humplik, A. Galashov, L. Hasenclever, P. A. Ortega, Y. W. Teh, and N. Heess, "Meta reinforcement learning as task inference," 2019, *arXiv:1905.06424*.
[34] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teho, "Set Transformer: A framework for attention-based permutation-invariant neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 97, Jun. 2019, pp. 3744–3753.
[35] B. Peng et al., "FACMAC: Factored multi-agent centralised policy gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Jan. 2020, pp. 12208–12221.
[36] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30. Red Hook, NY, USA: Curran Associates, Jan. 2017, pp. 1–22.
[37] F. Christianos, L. Schäfer, and S. V. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., Red Hook, NY, USA: Curran Associates, 2020, pp. 10707–10717.
[38] J. K. Terry, B. Black, and M. Jayakumar. (2020). *Magent*. GitHub repository. [Online]. Available: https://github.com/Farama-Foundation/MAgent
[39] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative, multi-agent games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35. Red Hook, NY, USA: Curran Associates, Jan. 2021, pp. 24611–24624.
[40] S. A. H. Minoofam, A. Bastanfard, and M. R. Keyvanpour, "TRCLA: A transfer learning approach to reduce negative transfer for cellular learning automata," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 5, pp. 2480–2489, May 2023.

Zijie Zhao (Graduate Student Member, IEEE) received the B.S. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, and the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

His current research interests include multiagent reinforcement learning and multitask learning.

Yuqian Fu (Graduate Student Member, IEEE) received the B.S. degree in electronic information from Wuhan University, Wuhan, China, in 2022. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His current research interests include multiagent reinforcement learning, deep learning, and game AI.

Jiajun Chai (Graduate Student Member, IEEE) received the B.S. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing.

His current research interests include multiagent reinforcement learning, deep learning, and large language model.

Yuanheng Zhu (Senior Member, IEEE) received the B.S. degree in automation from Nanjing University, Nanjing, China, in 2010, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015.

He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include deep reinforcement learning, game theory, game intelligence, and multiagent learning.

Dongbin Zhao (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in material process engineering from Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively.

He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing. His current research interests include deep reinforcement learning, autonomous driving, game artificial intelligence, and robotics.