Multi-agent In-context Coordination via Decentralized Memory Retrieval

Tao Jiang[1,2], **Zichuan Lin**[3*], **Lihe Li**[1,2], **Yi-Chen Li**[1,2], **Cong Guan**[1,2], **Lei Yuan**[1,2], **Zongzhang Zhang**[1,2*], **Yang Yu**[1,2], **Deheng Ye**[3]

[1]National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
[3]Tencent, Shenzhen, China
{jiangt,lilh,liyc,guanc,yuanl}@lamda.nju.edu.cn, {zzzhang, yuy}@nju.edu.cn,
{zichuanlin, dericye}@tencent.com

Abstract

Large transformer models, trained on diverse datasets, have demonstrated impressive few-shot performance on previously unseen tasks without requiring parameter updates. This capability has also been explored in Reinforcement Learning (RL), where agents interact with the environment to retrieve context and maximize cumulative rewards, showcasing strong adaptability in complex settings. However, in cooperative Multi-Agent Reinforcement Learning (MARL), where agents must coordinate toward a shared goal, decentralized policy deployment can lead to mismatches in task alignment and reward assignment, limiting the efficiency of policy adaptation. To address this challenge, we introduce **M**ulti-**A**gent **I**n-**C**ontext **C**oordination via Decentralized Memory Retrieval (MAICC), a novel approach designed to enhance coordination by fast adaptation. Our method involves training a centralized embedding model to capture fine-grained trajectory representations, followed by decentralized models that approximate the centralized one to obtain team-level task information. Based on the learned embeddings, relevant trajectories are retrieved as context, which, combined with the agents' current sub-trajectories, inform decision-making. During decentralized execution, we introduce a novel memory mechanism that effectively balances test-time online data with offline memory. Based on the constructed memory, we propose a hybrid utility score that incorporates both individual- and team-level returns, ensuring credit assignment across agents. Extensive experiments on cooperative MARL benchmarks, including Level-Based Foraging (LBF) and SMAC (v1/v2), show that MAICC enables faster adaptation to unseen tasks compared to existing methods.

Code — https://github.com/LAMDA-RL/MAICC

1 Introduction

In-Context Learning (ICL) has emerged as a compelling paradigm for few-shot generalization, enabling models to tackle novel tasks by interpreting contextual cues without explicit retraining (Brown et al. 2020). This approach is epitomized by large language models, whose remarkable in-context abilities, unlocked through pretraining on vast web-scale corpora, have set new standards in natural language processing (Dong et al. 2024). The success of this paradigm has catalyzed a parallel pursuit within Reinforcement Learning (RL) to instill agents with similar on-the-fly policy adaptation capabilities (Moeini et al. 2025). To this end, the prevailing strategy reformulates RL as a sequence modeling problem (Chen et al. 2021): agents are trained on diverse trajectory datasets to internalize learning algorithms, allowing them to adapt to novel downstream tasks by conditioning on a few contextual examples (Laskin et al. 2023; Lee et al. 2023). This burgeoning field of In-Context Reinforcement Learning (ICRL) has shown notable promise, but its success has mainly been demonstrated in structured environments such as single-agent grid worlds and game-based tasks.

ICRL has demonstrated strong capabilities for fast adaptation in single-agent environments. Typically, these methods condition on in-context trajectories and maintain a memory that is continuously updated with new online experiences to inform decision-making. Despite its notable success, extending this paradigm to cooperative Multi-Agent Reinforcement Learning (MARL) scenarios presents significant challenges. Unlike the single-agent setting, where an agent aims to maximize its individual cumulative rewards, cooperative MARL requires multiple agents to collaborate towards a shared objective (Yuan et al. 2023). This collaborative nature introduces distinct challenges, particularly when deployed in a decentralized manner (Kraemer and Banerjee 2016). Firstly, decentralized execution confines each agent to its local observations, often leading to a biased or incomplete understanding of the overall task characteristics. Secondly, agents typically receive only a shared team-level reward, making it difficult to assess individual contributions. This ambiguity in credit assignment can lead to the "lazy agent" problem (Sunehag et al. 2018), where certain agents fail to learn effective policies and contribute meaningfully to the team's success. These twin challenges of partial observability and credit assignment critically undermine the efficacy of conventional ICRL approaches in the MARL setting. Therefore, given the proven adaptive capabilities of ICL, a method that enables efficient adaptation to unseen cooperative tasks in decentralized multi-agent settings is urgently needed.

To address the above objective, we propose **M**ulti-**A**gent **I**n-**C**ontext **C**oordination via Decentralized Memory Retrieval (MAICC), a framework designed for rapid team coor-

*Joint Corresponding Authors
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dination under Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) (Oliehoek and Amato 2016). Specifically, we train a single centralized embedding model to extract fine-grained trajectory representations, and multiple decentralized embedding models for decentralized execution that approximate the centralized model to obtain team-level task information. With these pretrained models, we retrieve relevant multi-agent trajectories to serve as in-context examples. These retrieved trajectories, combined with agents' current sub-trajectories, are used to guide and improve the decision-making process. During test time, we introduce a novel memory mechanism that efficiently balances an online replay buffer with a multi-task offline dataset for trajectory retrieval. Building upon this memory, we design a hybrid utility score that integrates both individual- and team-level returns, thereby enabling more accurate credit assignment across agents.

We evaluate our method on several standard cooperative MARL benchmarks, including Level-Based Foraging (LBF) (Papoudakis et al. 2021) and the StarCraft Multi-Agent Challenge (SMAC) v1 and v2 (Samvelyan et al. 2019; Ellis et al. 2023). Experimental results show that MAICC, equipped with efficient trajectory retrieval for ICL, enables significantly faster adaptation to unseen tasks compared to existing ICRL and multi-task MARL baselines. Additionally, visualizations of the learned trajectory embeddings verify the effectiveness of our embedding model design, capturing both individual- and team-level behavior patterns. Ablation studies further isolate and confirm the contribution of each key component in our framework. Together, these findings demonstrate MAICC's strong empirical performance, its ability to address the limitations of prior ICRL approaches, and its potential for broader deployment in complex multi-agent scenarios.

2 Related Work

In-context RL. By framing RL as a sequence modeling problem, Decision Transformer (DT) (Chen et al. 2021) can make decisions based on provided prompts (Xu et al. 2022). Subsequent studies scaled up model size and training data, enabling agents to exhibit ICL capabilities (Lee et al. 2022; Reed et al. 2022). Algorithm Distillation (Laskin et al. 2023) takes a significant step towards ICRL by utilizing historical trajectories. This enables agents to automatically improve their performance through trial and error, without updating their parameters. Agentic Transformer (AT) (Liu and Abbeel 2023) further demonstrates that cross-episodic contexts can help agents leverage hindsight, thus enabling performance improvement at test time (Huang et al. 2024a). Decision-Pretrained Transformer (DPT) (Lee et al. 2023) explores an alternative approach by predicting the optimal action given random historical trajectories and the current state. Subsequently, Retrieval-Augmented Decision Transformer (RADT) (Schmied et al. 2024) introduces retrieval augmentation into ICRL, utilizing a DT-based embedding model to select relevant historical trajectories and thereby further aid action prediction. However, these methods have only demonstrated effectiveness on single-agent tasks with simple interactions (Sridhar et al. 2025) and perform poorly on more complex decentralized cooperative tasks. In contrast, our approach achieves efficient trajectory retrieval tailored to the characteristics of Multi-Agent Systems (MASs), thereby facilitating collaborative adaptation to unseen tasks. To the best of our knowledge, our method is the first ICRL approach for Dec-POMDPs (Oliehoek and Amato 2016).

Cooperative multi-agent RL. Many real-world problems are large-scale and complex, rendering single-agent modeling inefficient and often impractical (Feng et al. 2025). These challenges are more effectively addressed within the MAS setting (Dorri, Kanhere, and Jurdak 2018), where MARL provides a robust framework for solution (Yuan et al. 2023). In cooperative MARL, where agents pursue shared objectives, significant progress has been made in domains such as video games (Li et al. 2025), domain calibration (Jiang et al. 2024), and financial trading (Huang et al. 2024b). A central challenge in cooperative MARL is partial observability due to decentralized execution. The Centralized Training with Decentralized Execution (CTDE) framework (Lowe et al. 2017) addresses this by propagating team-level information to individual agents during training, thereby enhancing coordination at execution. Another key issue is the absence of individual rewards, which leads to the "lazy agent" problem (Sunehag et al. 2018), where agents fail to improve their policies due to an inability to assess their own contributions. Actor-critic methods such as COMA (Foerster et al. 2018) mitigate this by introducing counterfactual baselines for policy updates, while value-based approaches like QMIX (Rashid et al. 2020) achieve implicit credit assignment by enforcing monotonicity in the value function (Son et al. 2019; Wang et al. 2020b,a). In this work, we address both challenges within the ICRL framework by incorporating corresponding modules, thereby enabling rapid adaptation to unseen cooperative tasks.

3 Background

3.1 Multi-Agent Reinforcement Learning

A cooperative multi-agent task is typically modeled as a Dec-POMDP (Oliehoek and Amato 2016), defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, R, \Omega, O, \mathcal{N}, H, \rho \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively; $\rho \in \Delta(\mathcal{S})$ is the initial state distribution where $\Delta(\mathcal{S})$ represents the set of probability distributions over the state space $\mathcal{S}$; $H \in \mathbb{N}$ is the episode horizon where $\mathbb{N}$ denotes the set of natural numbers. Each episode begins with an initial state s^0 sampled from ρ. At each time step h, given the global state $s^h \in \mathcal{S}$, each agent $j \in \mathcal{N} = \{1, 2, \cdots, n\}$ receives a local observation $o_j^h \in \Omega$ generated by the observation function $O(s^h, j)$, and selects an action $a_j^h \in \mathcal{A}$ according to its individual learnable policy $\pi_j(a_j^h|\tau_j^h)$. Here, τ_j^h denotes the trajectory $(o_j^0, a_j^0, \cdots, o_j^h)$. The joint action is denoted as $\mathbf{a}^h = (a_1^h, a_2^h, \cdots, a_n^h)$. The environment then transitions to the next state $s^{h+1} \sim \mathcal{T}(\cdot|s^h, \mathbf{a}^h)$ and provides a global reward $r^h = R(s^h, \mathbf{a}^h)$. The episode terminates when a predefined condition is met or after H steps. The objective is to optimize the joint policy $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$ to maximize the value function $V^{\mathcal{M}}(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{h=0}^{H-1} r^h\right]$.

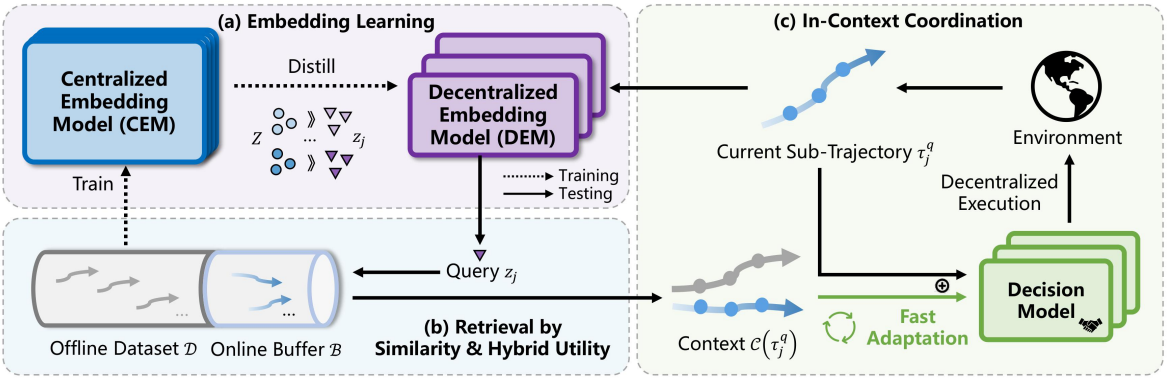


Figure 1: The conceptual workflow of MAICC. Dashed lines show data flow during centralized training, where CEM samples trajectories from the offline dataset for training and distills team information to DEMs. Solid lines show data flow during decentralized execution, where current sub-trajectories retrieve trajectories from the hybrid memory based on similarity and the hybrid utility score. Blue $\circ$ and purple $\triangledown$ denote different token embeddings output by CEM and DEMs, respectively. $\oplus$ denotes concatenation of the retrieved trajectories with the current sequence, which helps the decision models adapt quickly.

3.2 Decision Transformer

Transformers, originally developed for sequence modeling in language tasks (Vaswani et al. 2017), have been applied to RL by Decision Transformer (DT) (Chen et al. 2021), which frames decision-making as sequence modeling (Janner, Li, and Levine 2021). Instead of learning value functions as in traditional RL methods, DT derives policies from sequences of input tokens within a single trajectory, represented as $(\hat{R}^0, o^0, a^0, \hat{R}^1, o^1, a^1, \cdots)$, where each token corresponds to the Return-To-Go (RTG), observation, and action, respectively. The RTG at time step h, denoted as $\hat{R}^h$, is defined as the sum of future rewards: $\hat{R}^h = \sum_{t=h}^{H-1} r^t$. DT is trained in a supervised manner, similar to behavior cloning (BC) (Atkeson and Schaal 1997). During testing, by conditioning on a high RTG, DT can autoregressively generate actions aimed at achieving high cumulative rewards (return).

3.3 Problem Setting

In this paper, we focus on learning new cooperative tasks through limited online trial and error, without updating model parameters. This approach, known as ICRL (Moeini et al. 2025), is a practical instance of meta-RL (Beck et al. 2023). Formally, we consider tasks to be drawn from a distribution $P(\mathcal{M})$, where each task is a Dec-POMDP and the components of the tuple may differ across tasks. During training, agents have access only to datasets from a fixed set of tasks, denoted as $\mathcal{D} = \{\mathcal{D}_i\}$. Each dataset $\mathcal{D}_i$ contains multiple trajectories, collected from the corresponding task $\mathcal{M}_i$ by cooperative behavior policies $\boldsymbol{\mu}$, which is unknown to the learning agents. After pretraining, the model parameters are fixed. At test time, the agent team interacts with a new, unseen environment randomly sampled from $P(\mathcal{M})$ for only T episodes, with the goal of achieving fast coordination without parameter updating. This objective can be formulated as maximizing the expected return in the final adaptation episode under the task distributions: $\max \mathbb{E}_{\mathcal{M}\sim P(\cdot)} V^{\mathcal{M}}(\boldsymbol{\pi})$.

4 Method

In this section, we present the MAICC (**M**ulti-**A**gent **I**n-**C**ontext **C**oordination) framework, which exploits the ICL capabilities of Transformer-based models for rapid adaptation to unseen cooperative tasks. The overall architecture is shown in Fig. 1. During training, we first learn embedding models to capture the characteristics of multi-agent trajectories for efficient context retrieval (Sec. 4.1). Specifically, a centralized embedding model (CEM) extracts team-level information via autoregressive prediction, which guides decentralized embedding models (DEMs) for decentralized execution. The DEMs are then used to retrieve trajectories with similar embeddings for a given input, enabling the decision model to generate appropriate actions (Sec. 4.2). By leveraging these in-context trajectories, the pretrained decision model can infer task characteristics and generalize across diverse tasks. In the testing phase, we introduce a novel memory mechanism that combines an online replay buffer with offline datasets to enhance retrieval efficiency. We further propose a hybrid utility score that integrates individual- and team-level information to select high-quality in-context trajectories, promoting effective coordination (Sec. 4.3). Finally, we provide a theoretical analysis of the online cumulative regret of our approach (Sec. 4.4).

4.1 Multi-Agent Trajectory Embedding Models

Efficient multi-agent trajectory retrieval relies on learning high-quality trajectory embeddings. To achieve this, we adopt the CTDE paradigm (Kraemer and Banerjee 2016; Lowe et al. 2017) and design both centralized and decentralized embedding models. During training, agents have access to global team observations and actions, allowing the CEM to capture fine-grained team-level information. In contrast, during execution, each agent is limited to its own local observations and actions, resulting in less informative embeddings from the DEMs. To address this disparity, we employ the CEM to distill team-level knowledge into the DEMs dur-

ing training, thereby enhancing the DEMs' representational capacity for decentralized execution.

Formally speaking, we denote the number of agents as n. The multi-task offline dataset $\mathcal{D}$ consists of trajectories $\tau = (\boldsymbol{o}^0, \boldsymbol{a}^0, r^0, \ldots, \boldsymbol{o}^{H-1}, \boldsymbol{a}^{H-1}, r^{H-1})$, where $\boldsymbol{o} = (o_1, \ldots, o_n)$ and $\boldsymbol{a} = (a_1, \ldots, a_n)$. Our trajectory embedding models employ three types of tokens: observation o, action a, and post-step information $\hat{P}$. Following prior work (Liu and Abbeel 2023; Huang et al. 2024a), the token $\hat{P}$ comprises the global reward, done signal, and task completion flag, which are essential for modeling long-horizon trajectories. We omit the RTG token, as it can cause retrieval of trajectories from irrelevant tasks that happen to have similar RTG values, thereby reducing the informativeness of in-context examples and harming action prediction.

As illustrated in Fig. 2, the CEM receives the agents' local observations $\{o_j^h\}_{j=1}^n$, actions $\{a_j^h\}_{j=1}^n$, and post-step information $\hat{P}^h$ at each time step h, and outputs the corresponding embeddings: $\{Z_{o,j}^h\}_{j=1}^n, \{Z_{a,j}^h\}_{j=1}^n, Z_p^h = \mathrm{CEM}(\{o_j^h\}_{j=1}^n, \{a_j^h\}_{j=1}^n, \hat{P}^h)$. To be compatible with centralized training, we adapt the causal transformer by introducing intra-team visibility, enabling observation and action tokens within the same team and time step to attend to each other. We further design three loss functions to model the behavior policy ($\mathcal{L}_\mu$), reward function ($\mathcal{L}_R$), and observation transition dynamics ($\mathcal{L}_\mathcal{T}$) of the trajectory:

$$\mathcal{L}_{\mathrm{CEM}} = \mathcal{L}_\mu + \mathcal{L}_R + \mathcal{L}_\mathcal{T}, \tag{1}$$

$$\mathcal{L}_\mu = -\mathbb{E}_{\tau\sim\mathcal{D}} \sum_{h=0}^{H-1} \sum_{j=1}^{n} \log \mathrm{MLP}_{o\to a}(a_j^h | Z_{o,j}^h), \tag{2}$$

$$\mathcal{L}_R = \mathbb{E}_{\tau\sim\mathcal{D}} \sum_{h=0}^{H-1} \left(r^h - \sum_{j=1}^{n} \mathrm{MLP}_{a\to r}(Z_{a,j}^h) \right)^2, \tag{3}$$

$$\mathcal{L}_\mathcal{T} = -\mathbb{E}_{\tau\sim\mathcal{D}} \sum_{h=0}^{H-2} \sum_{j=1}^{n} \log \mathrm{MLP}_{p\to o}(o_j^{h+1} | Z_p^h, o_j^h), \tag{4}$$

where MLPs with different subscripts fit different functions. Eq. 3 can be regarded as performing implicit credit assignment (Sunehag et al. 2018; Rashid et al. 2020), which benefits subsequent decentralized adaptation.

During decentralized execution, the DEMs capture embeddings using only local information, i.e., $z_{o,j}^h, z_{a,j}^h, z_p^h = \mathrm{DEM}(o_j^h, a_j^h, \hat{P}^h)$. To enhance coordination, we introduce auxiliary objectives that distill team-level information by minimizing the KL divergence between the embeddings generated by the CEM and those produced by the DEMs:

$$\mathcal{L}_{\mathrm{DEM}} = \mathbb{E}_{\tau\sim\mathcal{D}} \sum_{h=0}^{H-1} \sum_{j=1}^{n} \left[\mathrm{KL}(Z_{o,j}^h, z_{o,j}^h) + \mathrm{KL}(Z_{a,j}^h, z_{a,j}^h) \right] + \mathbb{E}_{\tau\sim\mathcal{D}} \sum_{h=0}^{H-1} \mathrm{KL}(Z_p^h, z_p^h), \tag{5}$$

where $\mathrm{KL}(p, q)$ measures the divergence from the target distribution p to the approximate distribution q.

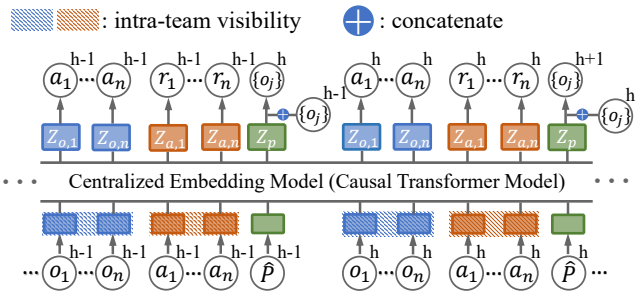


Figure 2: The illustration of CEM. Intra-team visibility enables observation and action tokens within the same team to attend to each other at each time step. The causal transformer predicts individual actions and rewards, while the post-step information token, concatenated with the previous individual observation, is used to predict the next observation.

4.2 Retrieval-Based In-Context Decision Training

To address diverse cooperative tasks with a single decision model, we use the trained DEMs to retrieve trajectories that inform action generation. Given an individual query sub-trajectory $\tau_j^q = (o_j^0, a_j^0, r^0, \cdots, a_j^{q-1}, r^{q-1}, o_j^q) \sim \mathcal{D}$ up to a certain time step, we first input it into the DEM and extract the embeddings at the final step, which, due to the transformer's long-range dependency modeling, summarizes the entire sub-trajectory. We then apply average pooling on the extracted embeddings over different tokens to obtain the final query embedding, i.e., $z_j^q = \mathrm{MEAN}(z_{o,j}^q, z_{a,j}^{q-1}, z_p^{q-1})$. Using Maximum Inner Product Search (MIPS) (Douze et al. 2024), we retrieve the top-k most relevant in-context trajectories: $\mathcal{C}(\tau_j^q) = \arg\max_{\tau^c\in\mathcal{D}}^k \mathrm{cossim}(z^c, z_j^q)$, where k is the number of in-context trajectories, cossim denotes cosine similarity, and z^c is the candidate embedding computed in the same manner as the query.

The retrieved in-context trajectories provide additional task-specific information to the current sub-trajectory. We concatenate these trajectories with the query and train the decision model π_θ (a causal transformer with parameter θ sharing across agents) using the following loss function:

$$\mathcal{L}_\pi = -\mathbb{E}_{\tau_j^q\sim\mathcal{D}} \log \pi_\theta \left(a_j^q | \mathrm{CONCAT}(\mathcal{C}(\tau_j^q), \tau_j^q) \right), \tag{6}$$

where CONCAT denotes the concatenate function. It is worth noting that, in addition to the three types of tokens—observation, action, and post-step information—the decision model also receives a RTG token. Unlike the embedding models, the decision model leverages the RTG from the retrieved trajectory to guide action selection towards achieving the desired return.

4.3 Decentralized In-Context Fast Coordination

After pretraining the embedding and decision models, a new task is randomly sampled from the task distribution $P(\mathcal{M})$. The agent team must then rapidly adapt and coordinate on this task without further parameter updates. During T episodes of interaction, data are stored in an online replay buffer. Agents can retrieve trajectories from both the

multi-task offline dataset $\mathcal{D}$, which may exhibit distribution shift, and the online buffer $\mathcal{B}$, which is aligned with the current task but initially contains limited experience. To address this, we propose a selective memory mechanism with exponential time decay: early episodes prioritize offline data to encourage exploration, while later episodes increasingly leverage high-value online trajectories to enhance exploitation. Specifically, we introduce a coefficient $\beta_t = \exp\left(-\lambda \frac{t}{T}\right)$ for episode t, where the hyper-parameter λ controls the decay rate (Ross, Gordon, and Bagnell 2011). We construct a new buffer $\mathcal{B}'$ by sampling from $\mathcal{D}$ with probability β_t and from $\mathcal{B}$ with probability $1 - \beta_t$. This method is simple, effective, and theoretically grounded.

Based on the constructed memory $\mathcal{B}'$, we further enhance the exploitation of high-value trajectories by introducing a hybrid utility score during inference, defined as $\mathcal{S}_{\text{util}}(\tau) = \alpha \,\text{norm}(\mathcal{R}) + (1-\alpha)\,\text{norm}(\tilde{\mathcal{R}})$. Here, $\mathcal{R} = \sum_{h=0}^{H-1} r^h$ is the global return, $\tilde{\mathcal{R}} = \sum_{h=0}^{H-1} \tilde{r}_j^h$ is the predicted individual return for agent j, $\text{norm}(\cdot)$ denotes normalization to $[0, 1]$, and $\alpha \in [0, 1]$ is a hyper-parameter. In Dec-POMDPs, where individual rewards are unavailable, we leverage the pretrained DEMs to predict individual rewards from action embeddings, i.e., $\tilde{r}_j^h = \text{MLP}_{a \to r}(z_{a,j}^h)$. This hybrid utility score enables agents to retrieve trajectories that are beneficial at both the individual and team levels, thereby mitigating the "lazy agent" problem in multi-agent systems. Incorporating the similarity score used during training, the retrieval process is formulated as $\mathcal{C}(\tau_j^q) = \arg\max_{\tau^c}^k \mathcal{S}(\tau^c, \tau_j^q)$, where $\tau^c \in \mathcal{B}'$ and $\mathcal{S}(\tau^c, \tau_j^q) = \text{cossim}(z^c, z_j^q) + \mathcal{S}_{\text{util}}(\tau^c)$. The decision model then outputs actions conditioned on the concatenation of the retrieved in-context trajectories and the input trajectory: $a \sim \pi_\theta(\cdot \,|\, \text{CONCAT}(\mathcal{C}(\tau_j^q), \tau_j^q))$. The overall pseudo code of MAICC is provided in Alg. 1.

Algorithm 1: Multi-agent In-context Coordination via Decentralized Memory Retrieval

Input: Initialized two trajectory embedding models CEM, DEM, decision model π_θ, multi-task offline dataset $\mathcal{D}$, empty online replay buffer $\mathcal{B}$

// Multi-Agent Trajectory Embedding Models
while not converged **do**
  Update CEM and DEM by Eq. 1 and Eq. 5 on $\mathcal{D}$
end while
// Retrieval-Based In-Context Decision Training
while not converged **do**
  Retrieve in-context trajectories $\mathcal{C}$ with DEM
  Update π_θ with $\mathcal{C}$ by Eq. 6
end while
// Decentralized In-Context Fast Adaptation
for $t = 1, 2, \cdots, T$ **do**
  Construct the memory $\mathcal{B}'$
  while episode not ended **do**
    Retrieve in-context trajectories $\mathcal{C}$ with $\mathcal{S}$ and $\mathcal{B}'$
    Decentralized execution with π_θ conditioned on $\mathcal{C}$
  end while
  Store episode trajectory in $\mathcal{B}$
end for

4.4 Theoretical Analysis

In this section, we provide a bound on the online cumulative regret of MAICC. For a given task $\mathcal{M}$ with $|\Omega| = \omega$, $|\mathcal{A}| = A$, and horizon H, let π^* denote the expert policy. The cumulative regret over T episodes is defined as $\mathbf{Reg}_{\mathcal{M}} = \sum_{t=1}^{T} V^{\mathcal{M}}(\pi^*) - V^{\mathcal{M}}(\hat{\pi}_t)$, where $\hat{\pi}_t = \beta_t \pi^{\mathcal{D}} + (1-\beta_t)\pi_t^{\mathcal{B}}$ with subscript t is the MAICC policy in episode t. Here, $\pi^{\mathcal{D}}$ is the policy that retrieves the in-context trajectories from the offline dataset $\mathcal{D}$, while $\pi_t^{\mathcal{B}}$ retrieves them from the online buffer $\mathcal{B}$ accumulated up to episode t, described in Sec. 4.3.

Assumption 1. ***(Sufficiency of Retrieval)*** *Let $\pi_t^{\mathcal{B}*}$ denote the policy that, for each query τ^q, directly uses the entire online buffer accumulated over t episodes as the in-context input (i.e., without retrieval). For all $(\tau^q, \mathcal{B}, t)$, we have $\pi_t^{\mathcal{B}}(a|\tau^q) = \pi_t^{\mathcal{B}*}(a|\tau^q)$ for all $a \in \mathcal{A}$.*

This assumption is trivially satisfied if the number of retrieved trajectories k equals t. Even when $k < t$, a carefully selected k trajectories can still capture most of the relevant information. Since Transformer inference time scales quadratically with context length, using a representative subset rather than the entire buffer is both efficient and practical.

Theorem 1. *Suppose $\sup_{\mathcal{M}} P(\mathcal{M})/P_{\mathcal{D}}(\mathcal{M}) \leq C$ for some $C > 0$, where $P_{\mathcal{D}}(\mathcal{M})$ denotes the training task distribution. Then the expected online cumulative regret of MAICC satisfies $\mathbb{E}_{P(\mathcal{M})}[\mathbf{Reg}_{\mathcal{M}}] \leq \tilde{\mathcal{O}}(CH^{3/2}\omega\sqrt{AT})$.*

MAICC offers a theoretical guarantee similar to prior ICRL methods (Schmied et al. 2024; Lee et al. 2023; Jing et al. 2024) as $\tilde{\mathcal{O}}$ is Big-O ignoring poly-logarithmic factors in complexity. In practice, however, the initial online replay buffer may lack sufficiently informative trajectories, leading to inefficient exploration. By leveraging our selective memory mechanism, MAICC adapts to new tasks more efficiently. Experimental results further support this advantage, and detailed derivations are provided in Appendix[1] B.

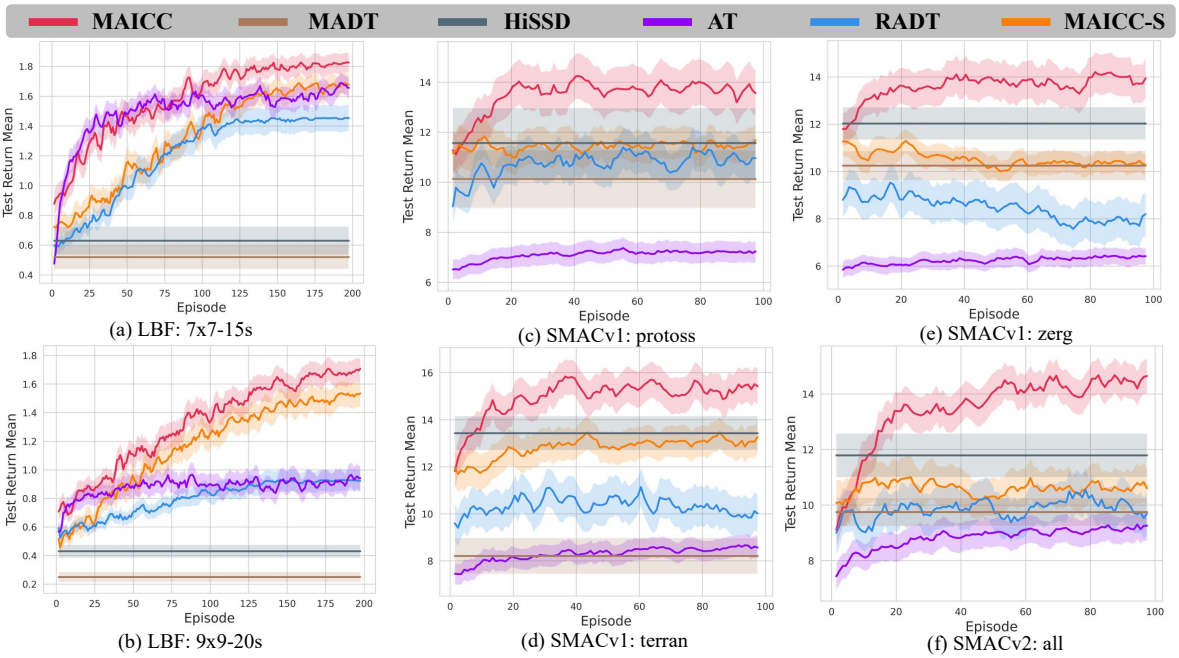
5 Experiments

In this section, we evaluate the proposed MAICC framework empirically. We begin by describing the experimental environments and baseline methods in Sec. 5.1. We then conduct a series of experiments to address the following questions: (1) How does MAICC compare to various baselines in terms of fast coordination (Sec. 5.2)? (2) How effectively do the DEMs capture representations of multi-agent trajectories (Sec. 5.3)? (3) What is the contribution of each component of MAICC to overall performance (Sec. 5.4)?

5.1 Experiment Setup

We evaluate MAICC on several cooperative benchmarks. The first is the **Level-Based Foraging (LBF)** (Papoudakis et al. 2021), a grid-world where agents must coordinate to collect food simultaneously. Each agent observes only a local field of view and collect food at different locations within a limited number of time steps. We consider two setups: *LBF: 7x7-15s* and *LBF: 9x9-20s*. We also evaluate on the

[1] Available in the code repository



(a) LBF: 7x7-15s (b) LBF: 9x9-20s (c) SMACv1: protoss (d) SMACv1: terran (e) SMACv1: zerg (f) SMACv2: all

Figure 3: In-context adaptation performance across different scenarios. Each scenario is evaluated over 50 test runs on randomly sampled tasks, with results reported as the mean return and 95% confidence interval.

StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019), using *Protoss*, *Terran*, and *Zerg* tasks where allied units cooperate to defeat enemy units controlled by the built-in AI. Tasks vary in agent types, numbers, and enemy configurations. Additionally, we test on the **StarCraft Multi-Agent Challenge-v2 (SMACv2)** (Ellis et al. 2023), an SMAC extension with increased randomness; for this benchmark we pretrain a single model to handle *all* three task types. For each scenario we use QMIX (Rashid et al. 2020) to collect the multi-task offline dataset $\mathcal{D}$. Further details are provided in Appendix C.

MAICC is pretrained on a multi-task offline dataset and adapts online in a decentralized manner for rapid coordination. For comparison, we include several baselines. MADT (Meng et al. 2023) extends DT to multi-agent settings and performs well on single tasks. AT (Liu and Abbeel 2023) and RADT (Schmied et al. 2024) are state-of-the-art ICRL methods trained offline for online adaptation, but they lack mechanisms specific to multi-agent coordination. HiSSD (Liu et al. 2025) learns generalizable skills from multi-task offline data but does not support online adaptation. MAICC-S is an ablated version of our method, where only the DEMs are trained for trajectory modeling during pretraining, without the CEM; all other components remain unchanged. Except for HiSSD, all methods are Transformer-based, and we use the same-size GPT-2 model (Radford et al. 2019) for fair comparison.

Experimental results are obtained by training each model with 5 different random seeds. For each seed, performance is evaluated on 10 random tasks, yielding a total of 50 test runs. We report the mean and 95% confidence intervals. Additional implementation details are provided in Appendix D.

5.2 Main Results

We first assess the in-context adaptation performance of our method and baselines across six test scenarios. As Fig. 3 shows, agent teams are required to increase their average return over the task distribution within a limited number of episodes per scenario. Our method consistently outperforms the baselines, enabling faster adaptation to unseen cooperative tasks without parameter updating.

Since MADT and HiSSD do not support online adaptation, their results are shown as fixed horizontal lines. On SMAC tasks their performance matches that of ICRL baselines; however, on LBF tasks—where agent observability is reduced—their performance deteriorates markedly, emphasizing the importance of in-context adaptation. AT predicts actions from trajectories of previous episodes and achieves good results only on the *LBF: 7x7-15s*. RA-DT likewise uses trajectory retrieval, but its coarse-grained embedding and absence of adaptation for cooperative scenarios limit its effectiveness. The performance gap between MAICC-S and our method further demonstrates the necessity of explicitly modeling multi-agent characteristics in trajectory embeddings. Notably, only our method exhibits clear in-context adaptation in complex SMAC scenarios; the gap is largest in *SMACv2: all*, which has the greatest task diversity, highlighting the promise of MAICC in large-scale data settings.

5.3 Visualization of Learned Embeddings

To evaluate the trajectory embedding model, we visualize embeddings for the *SMACv2: all* scenario. As shown in Fig. 4, trajectories are embedded and projected to two di-

Variants	EM With RTG	Coefficient β	CEM loss	Hyper-parameter α	*SMACv2: all* Ret.
Default	False	$\beta_t = \exp(-\lambda \frac{t}{T})$	$\mathcal{L}_\mu + \mathcal{L}_R + \mathcal{L}_\mathcal{T}$	$\alpha = 0.8$	14.51±0.46
(A)	True				13.52±0.62
(B)		$\beta_t = 0$			12.16±0.72
		$\beta_t = 1$			11.17±0.64
(C)			$\mathcal{L}_\mu + \mathcal{L}_R$		13.43±0.51
			$\mathcal{L}_\mu + \mathcal{L}_\mathcal{T}$	$\alpha = 1$	12.32±0.48
			$\mathcal{L}_\mu$	$\alpha = 1$	10.55±0.39
(D)				$\alpha = 1$	13.61±0.40
				$\alpha = 0$	13.26±0.66

Table 1: Ablation Study on MAICC. Unless otherwise noted, all settings follow the default configuration. "Ret." indicates the average return over 50 test runs (with 95% confidence interval), evaluated in the final adaptation episode.

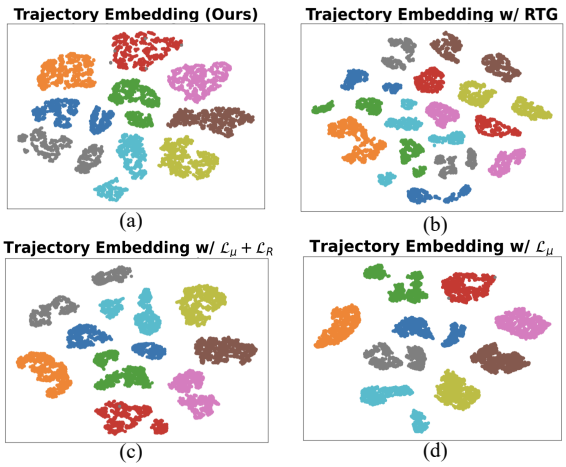


Figure 4: Embedding visualization for different training settings. Each point represents a trajectory embedding, and points with the same color belong to the same task.

mensions with t-SNE (Maaten and Hinton 2008).

We evaluate four embedding configurations. In our proposed setting (Fig. 4(a)), the embedding models are trained without the RTG token and utilize three loss terms, yielding fine-grained embeddings that cluster trajectories from the same task. Adding the RTG token (Fig. 4(b)) produces several small, overlapping clusters across tasks, increasing the risk of retrieving irrelevant trajectories. Fig. 4(c) and (d) use only a subset of the loss terms (Schmied et al. 2024), which induces coarse-grained, overly compact clusters—a sign of overfitting. Such representations generalize poorly to unseen tasks and hinder extrapolation. These results underscore the need to design embedding models and their loss functions carefully for effective trajectory retrieval.

5.4 Ablation Study

We evaluated the importance of different MAICC components by systematically modifying the default model and measuring performance changes on the most challenging scenario, *SMACv2: all*, as shown in Tab. 1.

In row (A), we examine the effect of incorporating the RTG token during embedding model training. The results show degraded performance, likely due to an increased likelihood of retrieving irrelevant trajectories.

Row (B) explores different values of β for memory construction. When the memory consists solely of either the offline dataset or the online buffer—instead of combining both sources using exponential time decay as coefficient—performance drops significantly. This indicates that each data source has limitations, and their weighted combination is crucial for effective adaptation to unseen tasks.

In row (C), we analyze the impact of different CEM loss functions. The results indicate that all three objectives are necessary; fine-grained trajectory modeling enhances action prediction. Notably, omitting $\mathcal{L}_R$ prevents individual return prediction during testing, further reducing the performance.

Row (D) evaluates the role of the hybrid utility score. Using only the global return ($\alpha = 1$) leads to insufficient credit assignment, while relying solely on the predicted individual return ($\alpha = 0$) may suffer from prediction inaccuracies. Therefore, the hybrid approach, which combines both, yields improved adaptation performance.

6 Conclusion and Discussion

In this paper, we address rapid cooperative adaptation in Dec-POMDPs by proposing MAICC, which enables agent teams to coordinate quickly on unseen tasks without parameter updates. During training, MAICC uses the CEM to extract fine-grained multi-agent trajectory representations and trains the DEMs to optimize them for decentralized execution. Given a current sub-trajectory, agents retrieve and concatenate relevant trajectories via the DEMs to train decision models. At test time, each agent retrieves trajectories from a memory that combines an online buffer and offline data. Credit assignment is achieved by combining team- and individual-level returns. Experiments on cooperative MARL benchmarks show that MAICC enables rapid adaptation to novel tasks. A limitation is reliance on exponential time decay for memory construction; incorporating uncertainty-based metrics (Lockwood and Si 2022) could further improve generalization and real-world deployment.

Acknowledgements

This work is supported by the National Science Foundation of China (62276126, 62250069, 62495093, 62506159, and U24A20324), the Natural Science Foundation of Jiangsu (BK20241199, BK20243039), and the AI & AI for Science Project of Nanjing University.

References

Atkeson, C. G.; and Schaal, S. 1997. Robot learning from demonstration. In *International Conference on Machine Learning*, 12–20.

Beck, J.; Vuorio, R.; Liu, E. Z.; Xiong, Z.; Zintgraf, L.; Finn, C.; and Whiteson, S. 2023. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 1877–1901.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 15084–15097.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; Li, L.; and Sui, Z. 2024. A survey on in-context learning. In *Empirical Methods in Natural Language Processing*, 1107–1128.

Dorri, A.; Kanhere, S. S.; and Jurdak, R. 2018. Multi-agent systems: A survey. *IEEE Access*, 6: 28573–28593.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Ellis, B.; Cook, J.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J. N.; and Whiteson, S. 2023. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *NeurIPS Datasets and Benchmarks Track*.

Feng, Z.; Xue, R.; Yuan, L.; Yu, Y.; Ding, N.; Liu, M.; Gao, B.; Sun, J.; Zheng, X.; and Wang, G. 2025. Multi-agent embodied ai: Advances and future directions. *arXiv preprint arXiv:2505.05108*.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, 2974–2982.

Huang, S.; Hu, J.; Chen, H.; Sun, L.; and Yang, B. 2024a. In-context decision transformer: reinforcement learning via hierarchical chain-of-thought. In *International Conference on Machine Learning*, 19871–19885.

Huang, Y.; Zhou, C.; Cui, K.; and Lu, X. 2024b. A multi-agent reinforcement learning framework for optimizing financial trading strategies based on TimesNet. *Expert Systems with Applications*, 237: 121502.

Janner, M.; Li, Q.; and Levine, S. 2021. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 1273–1286.

Jiang, T.; Yuan, L.; Li, L.; Guan, C.; Zhang, Z.; and Yu, Y. 2024. Multi-agent domain calibration with a handful of offline data. In *Advances in Neural Information Processing Systems*, 69607–69636.

Jing, Y.; Li, K.; Liu, B.; Zang, Y.; Fu, H.; Fu, Q.; Xing, J.; and Cheng, J. 2024. Towards offline opponent modeling with in-context learning. In *International Conference on Learning Representations*.

Kraemer, L.; and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190: 82–94.

Laskin, M.; Wang, L.; Oh, J.; Parisotto, E.; Spencer, S.; Steigerwald, R.; Strouse, D.; Hansen, S. S.; Filos, A.; Brooks, E.; Gazeau, M.; Sahni, H.; Singh, S.; and Mnih, V. 2023. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*.

Lee, J.; Xie, A.; Pacchiano, A.; Chandak, Y.; Finn, C.; Nachum, O.; and Brunskill, E. 2023. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, 43057–43083.

Lee, K.; Nachum, O.; Yang, M.; Lee, L.; Freeman, D.; Guadarrama, S.; Fischer, I.; Xu, W.; Jang, E.; Michalewski, H.; and Mordatch, I. 2022. Multi-game decision transformers. In *Advances in Neural Information Processing Systems*, 27921–27936.

Li, Z.; Ji, Q.; Ling, X.; and Liu, Q. 2025. A comprehensive review of multi-agent reinforcement learning in video games. *Authorea Preprints*.

Liu, H.; and Abbeel, P. 2023. Emergent agentic transformer from chain of hindsight experience. In *International Conference on Machine Learning*, 21362–21374.

Liu, S.; Shu, Y.; Guo, C.; and Yang, B. 2025. Learning generalizable skills from offline multi-task data for multi-agent cooperation. In *International Conference on Learning Representations*.

Lockwood, O.; and Si, M. 2022. A review of uncertainty for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 155–162.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Meng, L.; Wen, M.; Le, C.; Li, X.; Xing, D.; Zhang, W.; Wen, Y.; Zhang, H.; Wang, J.; Yang, Y.; and Xu, B. 2023. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2): 233–248.

Moeini, A.; Wang, J.; Beck, J.; Blaser, E.; Whiteson, S.; Chandra, R.; and Zhang, S. 2025. A survey of in-context reinforcement learning. *arXiv preprint arXiv:2502.07978*.

Oliehoek, F. A.; and Amato, C. 2016. *A concise introduction to decentralized POMDPs*. Springer.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2021. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *NeurIPS Datasets and Benchmarks Track*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Reed, S. E.; Zolna, K.; Parisotto, E.; Colmenarejo, S. G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J. T.; Eccles, T.; Bruce, J.; Razavi, A.; Edwards, A.; Heess, N.; Chen, Y.; Hadsell, R.; Vinyals, O.; Bordbar, M.; and de Freitas, N. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 627–635.

Samvelyan, M.; Rashid, T.; Schroeder de Witt, C.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The StarCraft multi-agent challenge. In *International Conference on Autonomous Agents and MultiAgent Systems*, 2186–2188.

Schmied, T.; Paischer, F.; Patil, V.; Hofmarcher, M.; Pascanu, R.; and Hochreiter, S. 2024. Retrieval-augmented decision transformer: External memory for in-context rl. *arXiv preprint arXiv:2410.07071*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, 5887–5896.

Sridhar, K.; Dutta, S.; Jayaraman, D.; and Lee, I. 2025. REGENT: A retrieval-augmented generalist agent that can act in-context in new environments. In *International Conference on Learning Representations*.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V. F.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2018. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and MultiAgent Systems*, 2085–2087.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020a. QPLEX: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*.

Wang, Y.; Han, B.; Wang, T.; Dong, H.; and Zhang, C. 2020b. DOP: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*.

Xu, M.; Shen, Y.; Zhang, S.; Lu, Y.; Zhao, D.; Tenenbaum, J.; and Gan, C. 2022. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning*, 24631–24645.

Yuan, L.; Zhang, Z.; Li, L.; Guan, C.; and Yu, Y. 2023. A survey of progress on cooperative multi-agent reinforcement learning in open environment. *arXiv preprint arXiv:2312.01058*.