IMPROVING LLM COLLABORATION VIA MULTI-AGENT PREFERENCE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Several works have explored *multi-agent reinforcement learning* (MARL) in LLM collaboration. However, constructing reliable rewards is difficult in practice, as complete and accurate metrics are often unavailable and hard to aggregate. Preference learning provides an alternative by learning from comparative human or AI feedback. Yet, its extension to multi-agent systems remains underexplored. To address this gap, we formulate preference-based multi-agent systems (MAS) from decentralized and centralized collaboration perspectives. We also introduce a general *multi-agent preference learning* framework (MAPL) to solve these problems. MAPL allows iterative updates by comparing the current solution with decentralized or centralized solutions generated by various agents. We instantiate MAPL using *MARL from human feedback* (MARLHF) with a learned reward model and *multi-agent direct preference optimization* (MADPO). Experiments on collaborative writing, coding, tool use, and travel planning show that MAPL can improve collaboration quality and efficiency while approaching the performance of MARL with fixed, well-defined rewards. Within MAPL, MARLHF generally outperforms MADPO on most tasks but remains sensitive to data coverage, agent and comparator models, and the underlying MARL algorithms.

1 INTRODUCTION

Recent advances in agentic AI have transformed LLMs into autonomous agents that can solve complex tasks by interacting with environments (Yao et al., 2023; Shinn et al., 2023; Wang et al., 2024; OpenAI, 2023; Bai et al., 2023; DeepSeek-AI, 2024; Shao et al., 2024; Comanici et al., 2025).

As LLM agents become more diverse and specialized, many studies have explored multi-agent cooperation to tackle complex tasks that are difficult for a single agent. Multi-agent systems (MAS) can be organized in several ways. For example, agents can play different roles and communicate to collaborate in a centralized manner (Li et al., 2023; Qian et al., 2024; Du et al., 2024). Such coordination can also be directed by a central manager agent (Wu et al., 2024; Hong et al., 2024). Alternatively, agents can solve subtasks independently on different devices to achieve higher efficiency (Zhang et al., 2024; Liu et al., 2026a;b). In such decentralized MAS, each agent processes only its local context. This lowers the computational requirements for each device and allows agents to be deployed locally, which can also improve data privacy. Nevertheless, most general-purpose LLMs are not explicitly trained for cooperation and thus often struggle to coordinate effectively with other agents (Park et al., 2025; Cemri et al., 2025).

Multi-agent reinforcement learning (MARL) has been extensively studied to improve coordination among agents (Zhang et al., 2021; Albrecht et al., 2024; Amato, 2024). Building on these advances, recent studies applied MARL to optimize LLM interaction (Zhao et al., 2026; Liao et al., 2025; Zhang et al., 2026; Liu et al., 2026a;b; Wang et al., 2026; Chen et al., 2026). However, many approaches rely on hand-crafted rewards based on domain-specific metrics, which may be unavailable or fail to accurately capture the intended task objective (Zhu et al., 2024; Bui et al., 2025).

Preference learning provides an alternative for optimizing LLM agents by leveraging pairwise human feedback without complex reward engineering (Christiano et al., 2017; Ziegler et al., 2019; Rafailov et al., 2023; Meng et al., 2024). Conventional preference learning methods mostly rely on fixed offline datasets, whereas some online approaches collect fresh preferences during training to reduce the distribution mismatch between the data and the evolving policy (Guo et al., 2024; Xiong

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2024). Yet, extending these methods to LLM collaboration remains challenging. The vast policy space of MAS makes it difficult to collect preference data to adequately cover diverse cooperation schemes (Oliehoek & Amato, 2016; Pan et al., 2022; Yang et al., 2021; Shao et al., 2023). Also, the learned rewards may be inaccurate or high-variance when the training data is insufficient, which could undermine the subsequent policy optimization.

To fill the gap, we study preference-based optimization of LLM collaboration. We formulate decentralized collaboration as a *preference-based decentralized partially observable MDP* (**Pb-Dec-POMDP**) and centralized collaboration as a *preference-based multi-agent partially observable MDP* (**Pb-MPOMDP**), and highlight their unique challenges. We introduce *multi-agent preference learning* (**MAPL**), a general framework that iteratively updates the agents' policies by comparing their joint solutions with those by a comparator model. MAPL considers three paradigms based on whether the agents and comparator operate in a decentralized or centralized manner: *decentralized collaboration with a decentralized comparator* (Dec-Dec), *centralized collaboration with a centralized comparator* (Cen-Cen), and *decentralized collaboration with a centralized comparator* (Dec-Cen). We also instantiate MAPL with two approaches: two-stage *multi-agent reinforcement learning from human feedback* (**MARLHF**) and a direct approach, *multi-agent direct preference optimization* (**MADPO**).

Our experiments on collaborative writing, coding, tool-use, and travel planning tasks show that MAPL can achieve performance similar to MARL with well-specified rewards. MAPL effectively improves decentralized and centralized collaboration under Dec-Dec and Cen-Cen, whereas its Dec-Cen variants perform poorly in most domains due to information and realizability gaps between the comparator and agents. MARLHF generally outperforms MADPO across most domains, but can exhibit high variance when preference data is limited, the comparator policy differs substantially from the agents', or actor-critic methods are used for policy optimization.

2 PRELIMINARIES

BT model in preference learning. Preference learning optimizes policies from comparative feedback. The Bradley-Terry (BT) model is a commonly used model to parameterize pairwise preference probabilities (Bradley & Terry, 1952). Let $\mathcal{X}$ and $\mathcal{Y}$ denote the prompt and response spaces, and let $R^* : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be an oracle reward. For $x \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$,

$$\Pr(y \succ y' \mid x) = \frac{\exp(R^*(x,y))}{\exp(R^*(x,y)) + \exp(R^*(x,y'))} = \sigma(R^*(x,y) - R^*(x,y')), \tag{1}$$

where $\sigma(z) = 1/(1+\exp(-z))$. In practice, a preference learning method typically uses a dataset $\mathcal{D} = (x_j, y_j^w, y_j^l)_{j=1}^m$, where y_j^w is preferred to y_j^l under prompt x_j.

Two-stage RLHF approaches. RLHF aligns LLMs from human preference (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023). Classic RLHF approaches involve two stages (Christiano et al., 2017; Ziegler et al., 2019), where a reward model $\widetilde{R}_\psi$ is first trained from pairwise preferences, then the policy π_θ is optimized with KL regularization against a reference policy π_{ref},

$$\begin{aligned} \mathcal{L}_{\text{RM}}(\psi) &= -\mathbb{E}_{(x,y^w,y^l)\sim\mathcal{D}}\left[\log\sigma\left(\widetilde{R}_\psi(x,y^w) - \widetilde{R}_\psi(x,y^l)\right)\right], \\ \mathcal{L}_{\text{RL}}(\theta) &= -\mathbb{E}_{x\sim\varrho,\, y\sim\pi_\theta(\cdot|x)}\left[\widetilde{R}_\psi(x,y) - \eta\log\frac{\pi_\theta(y\mid x)}{\pi_{\text{ref}}(y\mid x)}\right]. \end{aligned} \tag{2}$$

Here, ϱ denotes the prompt distribution and $\eta > 0$ is the KL penalty coefficient. We use RLHF to refer specifically to this two-stage, reward-modeling approach in the following parts.

One-stage DPO approaches. The two-stage RLHF pipeline requires both reward-model training and RL optimization, which is sensitive to hyperparameters and introduces substantial computational overhead (Chaudhari et al., 2025). Rafailov et al. (2023) proposed an efficient approach, DPO, that directly optimizes the policy without explicitly fitting a reward model,

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^w,y^l)\sim\mathcal{D}}\left[\log\sigma\left(\eta\log\frac{\pi_\theta(y^w\mid x)}{\pi_{\text{ref}}(y^w\mid x)} - \eta\log\frac{\pi_\theta(y^l\mid x)}{\pi_{\text{ref}}(y^l\mid x)}\right)\right]. \tag{3}$$

Under the standard DPO derivation, this objective corresponds to the same KL-regularized preference optimization problem as RLHF, while in practice, their performance can differ, depending on tasks, model realizability, and optimization (Shi et al., 2026; Razin et al., 2026).

3 MAS WITH PREFERENCE FEEDBACK

We study cooperative *multi-agent systems* (MAS), where LLM agents collaborate to solve a class of tasks (Qian et al., 2024; Hong et al., 2024; Chen et al., 2026; Liu et al., 2026a;b; Zhao et al., 2026). While most work assumes scalar rewards, we consider MAS with preference feedback.

3.1 PREFERENCE-BASED DECENTRALIZED MAS

Decentralized MAS support efficient and scalable execution, as LLM agents can run inference in parallel across different nodes (Zhang et al., 2024; Liu et al., 2026a;b). In this setting, each task is specified by language prompts and presented to the agents. At each turn, every agent generates a response based on its local context, and the individual responses are aggregated into a solution. Users, tools, and other models in the system then provide new requirements, feedback, or constraints that form the prompts for the next turn. This interaction continues until the task is completed or the turn limit is reached. For each episode, two trajectories are sampled, and an annotator compares their returns, producing preferences for policy optimization.

We formulate this problem as a *preference-based decentralized partially observable Markov decision process* (Pb-Dec-POMDP), denoted as $\langle \mathcal{I}, \mathcal{S}, \{\mathcal{O}_i\}_{i\in\mathcal{I}}, \{\mathcal{A}_i\}_{i\in\mathcal{I}}, T, O, R^*, H, b_0, P_{R^*}\rangle$ (Oliehoek & Amato, 2016). Here, $\mathcal{I} = \{1, \dots, n\}$ is a set of n LLM agents, where each agent i follows an individual policy π_i, and all policies form a joint policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_n)$. At each turn t, the state $s_t = (s_t^{\text{sys}}, s_t^{\text{usr}}) \in \mathcal{S}$ consists of an accessible part from the system $s_t^{\text{sys}} \in \mathcal{S}^{\text{sys}}$ and a latent part from users $s_t^{\text{usr}} \in \mathcal{S}^{\text{usr}}$, with initial state distribution $b_0 \in \Delta(\mathcal{S})$. The underlying state cannot be directly observed by the agents. Each agent i receives a natural-language prompt as its local observation $o_{i,t} \in \mathcal{O}_i$, and all agents' observations form a joint observation $\mathbf{o}_t = \{o_{1,t}, \cdots, o_{n,t}\} \in \mathcal{O}$. The observation $o_{i,t}$ at turn t provides a partial and potentially noisy view of s_t. Each agent generates a response as an action $a_{i,t} \sim \pi_i(\cdot|h_{i,t}), a_{i,t} \in \mathcal{A}_i$, which depends on its observation-action history $h_{i,t} = \{o_{i,0}, a_{i,0}, \dots, a_{i,t-1}, o_{i,t}\}$. The responses of all agents form the joint action $\mathbf{a}_t = \{a_{1,t}, \dots, a_{n,t}\} \in \mathcal{A}$, which induces a transition in the environment $s_{t+1} \sim T(\cdot|s_t, \mathbf{a}_t)$. The observation is then emitted as $\mathbf{o}_{t+1} \sim O(\cdot|s_{t+1}, \mathbf{a}_t)$ at the next turn $t+1$, where each agent receives its local observation $o_{i,t+1}$. We use H to denote the finite episode horizon.

In each episode of the Pb-Dec-POMDP, two trajectories $\tau, \tau' \in \Upsilon$ are sampled, and a human or AI annotator labels their preference according to $P_{R^*} : \Upsilon \times \Upsilon \to \{0, 1\}$, where $\Upsilon = \times_H(\mathcal{S} \times \mathcal{A})$ denotes the state-action trajectory space and $P_{R^*}(\tau, \tau')$ indicates whether τ is preferred to τ' under the latent oracle reward $R^* : \mathcal{S}^{\text{sys}} \times \mathcal{A} \to \mathbb{R}$. This yields a preferred trajectory τ^w and a dispreferred one τ^l. The oracle reward R^* is available to the annotator. Agents learn from the preferences over trajectories. The objective of the Pb-Dec-POMDP is to maximize the expected cumulative latent reward, $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}} \mathbb{E}_{(s_t, \mathbf{a}_t)\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{H-1} R^*(s_t^{\text{sys}}, \mathbf{a}_t)\right]$.

3.2 PREFERENCE-BASED CENTRALIZED MAS

In addition to decentralized collaboration, a MAS can also coordinate the agents through centralized communication or control. Agents can exchange information from their local histories to construct a shared context, which each agent then acts on (Wu et al., 2024; Chen et al., 2024). A manager agent can also aggregate available context about a task and decompose it into jobs for multiple workers (Wu et al., 2024; Hong et al., 2024). Such coordination helps agents avoid redundant work or missed subtasks, improving the quality of the joint solution.

This setting can be formulated as a *preference-based multi-agent POMDP* (Pb-MPOMDP), which follows the same tuple as the Pb-Dec-POMDP but adopts a centralized information structure (Messias et al., 2011). Specifically, in Pb-MPOMDP, the policies are conditioned on the full joint action-observation history $\mathbf{h}_t = \{\mathbf{o}_0, \mathbf{a}_0, \dots, \mathbf{a}_{t-1}, \mathbf{o}_t\}$. Each communicative agent i generates its response as $a_{i,t} \sim \pi_i(\cdot|\mathbf{h}_t)$ or a coordinator generates the joint response as $\mathbf{a}_t \sim \boldsymbol{\pi}_{\text{cen}}(\cdot|\mathbf{h}_t)$. The environment transition, learning objective, and preference labeling remain unchanged.

3.3 CHALLENGES

Data coverage and density. As it is often difficult and costly to simulate environments involving humans, many preference learning methods adopt pre-collected preference datasets for policy optimization (Ziegler et al., 2019; Rafailov et al., 2023). However, such datasets may not adequately cover the trajectories induced by the policies, and thus provide insufficient comparisons for effective training (Zhan et al., 2024). This limitation is amplified in MAS, which induces a combinatorially large policy space (Oliehoek & Amato, 2016; Yang et al., 2021; Shao et al., 2023). The lack of data coverage and density can limit effective optimization, as preferences collected from sparse or poorly covered regions may fail to guide learning toward high-quality cooperation policies (Zhan et al., 2024; Pan et al., 2022; Barde et al., 2024). MAPL supports online preference collection, where diverse trajectory comparisons are obtained throughout training.

Multi-agent preference datasets. Although there exist many preference datasets for single LLM alignment (Nakano et al., 2021; Bai et al., 2022; Cui et al., 2024; Köpf et al., 2023), standard datasets tailored to multi-agent collaboration are very limited. An intuitive approach is to manually adapt existing datasets by assigning roles or subtasks to individual agents. However, many of these tasks cannot be divided into subtasks meaningfully (Liu et al., 2026b). Even when such a division is possible, it remains unclear whether agents can achieve the optimal (or a good) solution under such task decomposition (Dietterich, 2000). MAPL uses joint trajectories of the agents being optimized in the targeted environment to build the dataset. Therefore, the trajectories in the dataset largely preserve the achievable cooperation of these agents.

Accuracy of reward estimates. In cooperative MAS, agents collaborate toward a shared team objective (Tan, 1993; Claus & Boutilier, 1998; Oliehoek & Amato, 2016; Samvelyan et al., 2019; Amato, 2024; Albrecht et al., 2024). Therefore, it is natural to assign preference to trajectories based on the joint policy. However, unlike MARL with fixed and well-specified rewards, preference-based RL must learn rewards from preferences (Christiano et al., 2017; Ziegler et al., 2019). With limited or noisy data, such reward estimates can be inaccurate or high-variance, making subsequent value estimation and policy updates less reliable. In addition, when the implicit reward model is updated alongside the policies (e.g., Rafailov et al., 2023; Guo et al., 2024), the training is non-stationary. MAPL maintains a dynamic replay buffer containing diverse samples from different stages of training. We also incorporate a group-relative Monte Carlo-based MARLHF approach (Equation 5) and use mini-batch stochastic gradient descent for policy optimization to improve training stability.

4 MULTI-AGENT PREFERENCE LEARNING

We introduce MAPL (Figure 1), a general framework for optimizing cooperative MAS from preference feedback. We instantiate MAPL with two representative approaches, MARLHF and MADPO, and discuss their key design features. Further details and additional MAPL variants are provided in Appendices A and B, respectively.

4.1 FRAMEWORK

Figure 1(b) illustrates the overall framework of MAPL. We initialize training by collecting the full joint histories $\mathbf{h}_H = \{\mathbf{o}_0, \mathbf{a}_0, \ldots, \mathbf{o}_H\}$ and the corresponding trajectories $\tau_H = \{s_0, \mathbf{a}_0, \ldots, s_H\}$ generated by the LLM agents for each task. For each pair of trajectories τ_H and τ'_H of a task, the annotator compares their returns following R^*, labels the trajectory with the higher return as the winner τ^w and the other as the loser τ^l, and discards ties. This yields an initial preference dataset $\mathcal{D}^{(0)} = \{(\mathbf{h}^w_H, \tau^w_H, \mathbf{h}^l_H, \tau^l_H)_j\}_{j=1}^m$ containing m preference pairs, with $\mathbf{h}^w_H$ and $\mathbf{h}^l_H$ denoting the joint histories associated with the winning and losing trajectories, respectively. We then select the top-k preference pairs according to the returns of their winning trajectories to construct the replay buffer $\mathcal{B}^{(0)} \subseteq \mathcal{D}^{(0)}$, which is used for the first iteration of MAPL training.

Since raw LLMs are not inherently cooperative, $\mathcal{D}^{(0)}$ may contain many trajectories with weak or no cooperation. As training progresses, the agents can generate higher-quality cooperative trajectories, providing more informative preference data for subsequent optimization. At each iteration ν, we collect m trajectories $\{(\mathbf{h}_H, \tau_H)_j\}_{j=1}^m \sim \boldsymbol{\pi}$ from the current agent policies and pair them with m trajectories $\{(\mathbf{h}'_H, \tau'_H)_j\}_{j=1}^m \sim \boldsymbol{\varpi}$ generated by a comparator policy, such as previous checkpoints

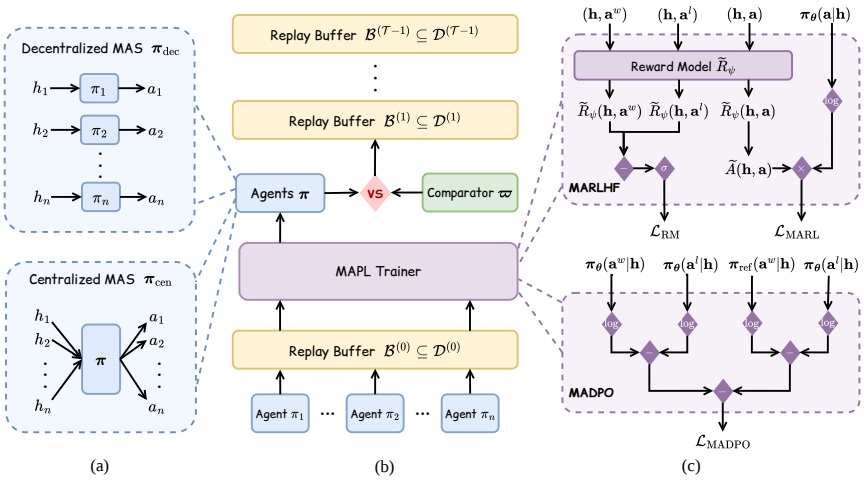

Figure 1: An illustration of the MAPL framework and methods. (a) Decentralized and centralized MAS; (b) MAPL framework; (c) Procedures of MARLHF and MADPO.

or other model variants. The annotator compares each pair to construct a new preference dataset $\mathcal{D}^{(\nu)}$, where the top-k preference pairs are selected to form the replay buffer $\mathcal{B}^{(\nu)} \subseteq \mathcal{D}^{(\nu)}$. This procedure is repeated until the maximum number of training iterations $\mathcal{T}$ is reached.

The agent policies and the comparator policy can adopt either decentralized or centralized collaboration, as illustrated in Figure 1(a). Under decentralized collaboration, each agent conditions only on its own local history without explicit communication, using $\pi_i(\cdot|h_i)$ or $\varpi_i(\cdot|h_i')$; under centralized collaboration, the policy conditions on the joint history, using $\boldsymbol{\pi}_{\text{cen}}(\cdot|\mathbf{h})$ or $\boldsymbol{\varpi}_{\text{cen}}(\cdot|\mathbf{h}')$. Since centralized policies have access to global information, they can potentially generate better trajectories with tighter coordination for training. Combining these choices, we therefore consider three training paradigms: *decentralized collaboration with a decentralized comparator* (Dec-Dec), *centralized collaboration with a centralized comparator* (Cen-Cen), and *decentralized collaboration with a centralized comparator* (Dec-Cen).

4.2 Methods

After obtaining $\mathcal{B}^{(\nu)}$ at iteration ν, we can either learn a reward model and optimize the policies with MARL, or directly optimize the policies from trajectory preferences.

MARLHF. In MARLHF, at each iteration ν, we first learn a history-based reward model $\widetilde{R}_\psi(\mathbf{h}_t, \mathbf{a}_t)$ to approximate the expected reward $\mathbb{E}[R^*(s_t^{\text{sys}}, \mathbf{a}_t)|\mathbf{h}_t, \mathbf{a}_t]$ for the given joint history. Specifically,

$$\mathcal{L}_{\text{RM}}(\psi) = -\mathbb{E}_{(\mathbf{h}_H^w, \mathbf{h}_H^l)\sim\mathcal{B}^{(\nu)}}\left[\log\sigma\left(\sum_{t=0}^{H-1}\left[\widetilde{R}_\psi(\mathbf{h}_t^w, \mathbf{a}_t^w) - \widetilde{R}_\psi(\mathbf{h}_t^l, \mathbf{a}_t^l)\right]\right)\right]. \tag{4}$$

The reward model is then held fixed during policy optimization. Although many MARL algorithms are available for the subsequent policy optimization (Albrecht et al., 2024; Amato, 2024), not all are well-suited to this setting. For example, due to the extremely large action space of LLM-based MAS, value-based methods are typically ineffective for LLM policy optimization. Moreover, in practice, the critic may be fit to an inaccurate reward model, especially in the early stage of training. These errors then propagate into the value estimates and undermine policy optimization.

As learning a critic is both difficult and costly, we use a group-based Monte Carlo method for MARL optimization (Liu et al., 2026b). For each task and policy update, we independently sample $G > 1$ trajectories $\tau_H^{\mathcal{G}} = \{\tau_H^{(1)}, \ldots, \tau_H^{(G)}\}$ and their corresponding joint histories $\mathbf{h}_H^{\mathcal{G}} = \{\mathbf{h}_H^{(1)}, \ldots, \mathbf{h}_H^{(G)}\}$ by executing the current joint policy $\boldsymbol{\pi}_{\text{old}}$. For compactness, we use $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{h}_H) = \prod_{t=0}^{H-1}\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{h}_t)$ to denote the product of joint action probabilities along the history $\mathbf{h}_H$. We adopt the averaged prediction within each group as a baseline to construct the group-relative advantage $\widetilde{A}^{(g)}$ and optimize

the agents according to,

$$\mathcal{L}_{\mathrm{MARL}^{\mathcal{G}}}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{h}_H^{\mathcal{G}} \sim \boldsymbol{\pi}_{\mathrm{old}}} \left[\frac{1}{G} \sum_{g=1}^{G} \mathsf{sg} \left[\frac{G}{G-1} \widetilde{A}^{(g)} - \eta \log \frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{h}_H^{(g)})}{\boldsymbol{\pi}_{\mathrm{ref}}(\mathbf{h}_H^{(g)})} \right] \log \boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{h}_H^{(g)}) \right]. \quad (5)$$

where $\widetilde{A}^{(g)} = \sum_{t=0}^{H-1} \left[\widetilde{R}_\psi(\mathbf{h}_t^{(g)}, \mathbf{a}_t^{(g)}) - \frac{1}{G} \sum_{g'=1}^{G} \widetilde{R}_\psi(\mathbf{h}_t^{(g')}, \mathbf{a}_t^{(g')}) \right]$, sg denotes the stop-gradient operator. $\frac{G}{G-1}$ corrects the group average, and the KL term includes the log-ratio as a fixed weight to produce the trajectory-level KL gradient in expectation. The agents' policies π_{θ_i} and $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ are then updated as $\nabla_{\theta_i} \mathcal{L}_{\mathrm{MARL}^{\mathcal{G}}}(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{MARL}^{\mathcal{G}}}(\boldsymbol{\theta})$ in decentralized and centralized MAS, respectively.

MADPO. We also present a more efficient approach that directly optimizes the policy from $\mathcal{B}^{(\nu)}$. At each iteration ν, MADPO minimizes

$$\mathcal{L}_{\mathrm{MADPO}}(\boldsymbol{\theta}) = -\mathbb{E}_{(\mathbf{h}_H^w, \mathbf{h}_H^l) \sim \mathcal{B}^{(\nu)}} \left[\log \sigma \left(\eta \log \frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{h}_H^w)}{\boldsymbol{\pi}_{\mathrm{ref}}(\mathbf{h}_H^w)} - \eta \log \frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{h}_H^l)}{\boldsymbol{\pi}_{\mathrm{ref}}(\mathbf{h}_H^l)} \right) \right]. \quad (6)$$

MADPO implicitly parameterizes the reward as $\widetilde{R}_{\boldsymbol{\theta}}(\mathbf{h}_t, \mathbf{a}_t) = \eta \log \frac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{h}_t)}{\boldsymbol{\pi}_{\mathrm{ref}}(\mathbf{a}_t|\mathbf{h}_t)}$. Then we can minimize the loss in Equation 6 with respect to each agent's parameters θ_i under decentralized collaboration, or the coordinator's parameters $\boldsymbol{\theta}$ under centralized coordination.

We provide further details about MAPL derivations and algorithms in Appendix A.

5 Experiments

We evaluate MAPL in four domains: `TLDR` for document processing (Völske et al., 2017), `CoopHE` for programming (Liu et al., 2026b), `BFCL` for function calling (Patil et al., 2025), and `Travel` for travel planning (Xie et al., 2024). Experimental settings, additional results, prompt design, compute resources, and our code are provided in Appendices C, D, E, F, and G, respectively.

5.1 Setup

As discussed in § 3, existing large-scale preference datasets (Bai et al., 2022; Cui et al., 2024) cannot be seamlessly repurposed for MAPL, since many tasks are atomic and cannot induce meaningful collaboration. Following prior work (Christiano et al., 2017; Lee et al., 2021; Kim et al., 2023; Zhu et al., 2024), we use deterministic oracle reward functions to generate preferences (Appendix C.3).

TLDR. Extracting key information from long context is an important ability for LLM agents. We instantiate this setting as a text summarization task built on the *TL;DR* dataset (Völske et al., 2017). In `TLDR`, two *Qwen3-1.7B* agents collaborate to summarize Reddit posts provided in the *prompt* field. One agent produces a concise high-level summary, while the other provides a detailed summary with more elaborated information. Their outputs are merged into a single briefing presented to the reader. We measure summarization quality with a weighted score over structural quality, stylistic consistency, and logical coherence, with details provided in Appendix C.3.

CoopHE. Prior work has shown the promise of employing multiple LLMs for software development (Talebirad & Nadiri, 2023; Qian et al., 2024). We take the collaborative code generation tasks in the *CoopHumanEval* (Liu et al., 2026b) dataset as a representative case. In `CoopHE`, two *Qwen2.5-Coder-3B* coding agents are solving basic *Python* programming problems, each specified by a natural-language prompt in *prompt* and a target function signature in *entry_point*. One agent implements utility modules in *aux*$(\cdot)$, while the other constructs *main*$(\cdot)$ by invoking them. The two functions are aggregated into a single file as a solution. We report the test pass rate for functional correctness and the valid auxiliary call rate for collaboration.

BFCL. Modern LLM agents interact with the systems through diverse tools (e.g., MCP servers, shell toolkits, and browser interfaces) (Anthropic, 2024; Yang et al., 2024; Zhou et al., 2024), yet sequentially invoking these tools incurs substantial latency. Multi-agent collaboration mitigates this by issuing function calls in parallel. We use the *parallel* and *parallel_multiple* tasks in the *Berkeley Function Calling Leaderboard* (`BFCL`) to demonstrate this concept (Patil et al., 2025). Two *Qwen3-4B-Instruct-2507* agents are jointly generating structured function calls for the request in

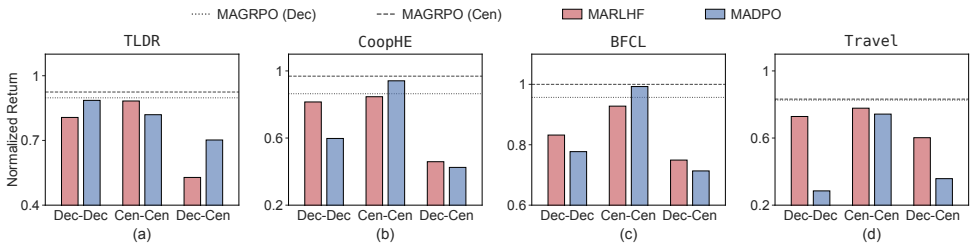

Figure 2: A comparison of MARLHF (red) and MADPO (blue) under Dec-Dec, Cen-Cen and Dec-Cen on `TLDR`, `CoopHE`, `BFCL`, and `Travel` across 5 runs. Dotted and dashed lines denote the performance of MAGRPO on decentralized and centralized MAS. Results are normalized to the reward scale w.r.t. each domain.

user_prompt, following the schemas in *function*. In *parallel* tasks, the agents share a single function schema but may call it with different arguments, whereas in *parallel_multiple* tasks, they must select the appropriate function from several available schemas. Agents are expected to learn a cooperation scheme that balances workloads and produces complementary calls. We use functional correctness and costs (#calls) as evaluation metrics.

Travel. We also consider a more concrete agentic domain, where two *Qwen3-4B-Instruct-2507* agents work together to plan a short trip. We draw tasks from *TravelPlanner* (Xie et al., 2024). We select the ones with *days* $\leqslant 5$ and *visiting_city_number* $\leqslant 2$ for simplicity and finer-grained comparison. The full request is provided in *query*, while candidate itinerary options and their associated costs are given in *reference_information*. In `Travel`, one agent arranges logistics (e.g., the city route, transportation, and accommodation), while the other designs activities (e.g., restaurants and attractions). Their outputs are combined into a single itinerary. We report a coarse-grained and a fine-grained metric: *pass* measures the proportion of plans satisfying all commonsense constraints (e.g., consistency between scheduled activities and the city occupied on that day), whereas *success* additionally requires compliance with the budget limits in *budget* and the user-specified constraints in *local_constraint*, including room types, food and transportation preferences.

Baselines. We compare our methods against three categories of baselines: single-agent methods, test-time LLM collaboration frameworks, and MARL approaches, with detailed prompts in Appendix E. Single-agent baselines employ a single LLM comparable in size to the MAS. We include the base model and its fine-tuned variants by RL and preference learning algorithms. For test-time collaboration, we consider three representative frameworks: *parallel*, where agents execute concurrently without communication; *sequential*, where one agent executes after another and communicates one-way (Wu et al., 2024); and *discussion*, where agents engage in bidirectional communication (Du et al., 2024). Given the prohibitively large policy space of LLMs, value-based methods are often impractical. We thus consider policy gradient methods as MARL baselines: multi-agent GRPO (MAGRPO) (Liu et al., 2026b), which extends GRPO to the multi-agent setting with group-relative advantages, and collaborative-LLM actor-critic with a centralized critic (CoLLM-CC) and with decentralized critics (CoLLM-DC) (Liu et al., 2026a).

5.2 Experimental Results

We compare MAPL with single-agent, test-time collaboration, and oracle-reward MARL baselines. We set $G = 4$ in Equation 5 and draw comparators from the same model family as the agents. The decentralized comparator uses the same model as each agent, whereas the centralized comparator approximately matches the agents' combined parameter count for a fair comparison (Appendix C).

As shown in Table 1 and Figure 2, MAPL under the best configurations can reach pass rates of 65.8% on `CoopHE` and 76.4% on `Travel`, exceeding the strongest single-agent baseline on `CoopHE` and `Travel` and outperforming test-time collaboration across all four domains. MARL methods attain the highest `TLDR` score and pass rates on `CoopHE` and `Travel`, since oracle rewards align policy optimization directly with the task objectives. Nevertheless, MAPL closely approaches MAGRPO, with less than 1% difference in `TLDR` score and `Travel` pass rate. Surprisingly, we find MARLHF can even surpass MAGRPO on `CoopHE` call rate (40.2% vs. 37.9%) and `Travel` success (13.1% vs. 12.5%) under decentralized collaboration. We attribute this to a higher density of samples of similar quality in the MAPL preference dataset (e.g., correct code with different cooperation patterns), which could provide richer local training signals for distinguishing their relative quality.

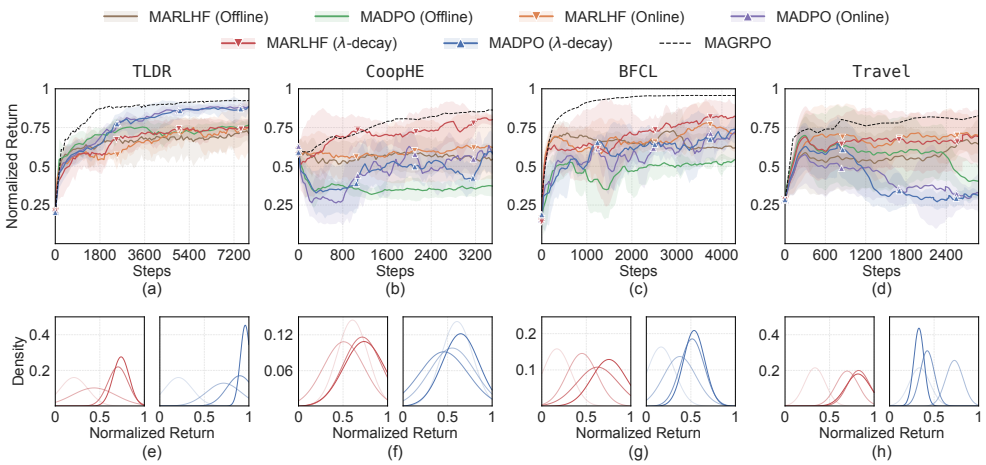

Figure 3: Evaluation results (a-d) and data distribution shift (e-h) of MARLHF and MADPO on offline (brown and green), online (orange and purple), λ-decay (red and blue) replay buffers. Curves are smoothed by an exponential moving average, and shaded regions indicate the standard deviation. Triangles mark the start of iterations. Data distributions are fitted with a Gaussian. Curves with higher opacity indicate later iterations.

Among the MAPL paradigms, Cen-Cen generally performs best for both MARLHF and MADPO across most domains, as centralized agents can exploit more complete information to coordinate. Dec-Dec also achieves strong performance after proper tuning and can outperform the raw larger model on `CoopHE`. This demonstrates the potential of decentralized collaboration, where multiple smaller models can cooperate to surpass a larger one. Cen-Cen underperforms Dec-Dec with MADPO on `TLDR`, where both the centralized agent and the comparator use a relatively small model compared with those in other domains. This suggests that centralized collaboration may require greater model capacity than individual decentralized agents, as centralized information typically induces a longer context for model reasoning. Dec-Cen generally performs worse due to the information and realizability gap between the comparator and agents. We find that this decrease is more salient in MARLHF than in MADPO, as its two-stage pipeline amplifies this representation error.

Method	TLDR	CoopHE		BFCL		Travel	
	Score	Pass	Call	Correct	Costs	Pass	Success
(a) Single Agent							
Raw Model	30.3	56.3	*N/A*	47.5	2.4	71.3	9.6
GRPO (Shao et al., 2024)	91.7	61.8	*N/A*	50.2	2.5	74.5	**25.0**
PPO (Schulman et al., 2017)	94.5	62.5	*N/A*	**56.3**	2.5	73.2	14.5
RLHF (Christiano et al., 2017)	79.4	65.0	*N/A*	48.3	2.3	75.6	5.6
DPO (Rafailov et al., 2023)	88.0	44.2	*N/A*	49.2	2.4	59.8	3.8
(b) Test-Time Collaboration							
Parallel (Wu et al., 2024)	22.9	50.0	18.4	12.8	1.2	66.2	8.3
Sequential (Wu et al., 2024)	21.7	62.5	24.6	25.5	1.3	66.9	10.0
Discussion (Du et al., 2024)	22.3	25.0	12.4	9.8	1.4	62.5	4.2
(c) MARL with R^*							
MAGRPO (Liu et al., 2026b)	93.5	71.3	37.9	43.8	1.0	76.2	12.5
CoLLM-DC (Liu et al., 2026a)	**95.4**	59.3	23.2	28.7	**0.9**	42.8	0.0
CoLLM-CC (Liu et al., 2026a)	95.2	**75.2**	39.7	53.4	1.1	**84.3**	**25.0**
(d) MAPL (Ours)							
MARLHF (Dec-Dec)†	86.3	61.0	40.2	22.7	1.0	70.5	13.1
MARLHF (Cen-Cen)	92.0	64.4	**46.8**	38.5	2.5	76.4	14.5
MARLHF (Dec-Cen)	56.4	20.3	14.3	14.9	1.3	62.4	8.3
MADPO (Dec-Dec)‡	92.9	58.8	28.5	15.6	1.1	41.5	0.0
MADPO (Cen-Cen)	87.4	65.8	44.9	48.3	2.5	73.6	12.5
MADPO (Dec-Cen)	74.9	16.5	12.2	11.1	1.0	40.6	4.2

Table 1: Evaluation results of MAPL and baseline methods on `TLDR`, `CoopHE`, `BFCL`, and `Travel` across 5 runs: Score measures the summary quality; Pass denotes the pass rate; Call denotes the invocation rate of *aux(·)*; Correct is the functional correctness rate; Costs measures the number of calls per agent; Pass is the commonsense pass rate; Success is the final pass rate with more constraints. Underline and **bold** indicate the best results across MAPL and all methods. † and ‡ denote pivots for the ablation studies in Table 2.

5.3 Ablations

We ablate MAPL on replay buffers, comparators, and MARLHF methods. We use the Dec-Dec paradigm as the representative setting and provide additional results in Appendix D.

We compare three ways of constructing the replay buffer. Offline MAPL uses 80 pre-collected preference pairs and selects the 16 highest-return samples. Online MAPL collects the same total amount of data over four iterations, where we append the 4 best samples from 20 preference pairs to the buffer at each iteration. We also consider a λ-decay replay buffer (the default in Table 1), which downweights samples from earlier iterations with ratio λ. We use tuned hyperparameter $\lambda = 0.8$ for MARLHF and $\lambda = 0.2$ for MADPO. Table 2(a) shows that online MAPL outperforms offline MAPL on `TLDR`, `CoopHE`, and `BFCL`. For example, MARLHF improves from 7.5% to 14.8%, and MADPO from 3.1% to 15.3% on `BFCL`. MAPL with λ-decay buffer achieves the best results across most domains. We find this improvement to be more pronounced for MARLHF than MADPO, since the broader data coverage supports learning a reward model with more diverse representations.

Method	TLDR	CoopHE	BFCL	Travel
	Score	Pass	Correct	Pass
(a) Replay Buffer				
MARLHF (λ-Decay)†	86.3	**61.0**	**22.7**	**70.5**
MARLHF (Offline)	70.9 (↓15.4)	46.4 (↓14.6)	7.5 (↓15.2)	63.6 (↓6.9)
MARLHF (Online)	76.0 (↓10.3)	59.7 (↓1.3)	14.8 (↓7.9)	69.8 (↓0.7)
MADPO (λ-Decay)‡	**92.9**	58.8	15.6	41.5
MADPO (Offline)	87.1 (↓5.8)	23.0 (↓35.8)	3.1 (↓12.5)	46.8 (↑5.3)
MADPO (Online)	88.7 (↓4.2)	58.5 (↓0.3)	15.3 (↓0.3)	43.2 (↑1.7)
(b) Comparator				
MARLHF (Same-size)†	86.3	61.0	22.7	70.5
MARLHF (2×)	66.9 (↓19.4)	48.8 (↓12.2)	18.8 (↓3.9)	56.4 (↓14.1)
MARLHF (4×)	66.5 (↓19.8)	50.0 (↓11.0)	**22.9** (↑0.2)	65.6 (↓4.9)
MARLHF (*Deepseek-Pro*)	70.1 (↓16.2)	63.5 (↑2.5)	21.8 (↓0.9)	**71.2** (↑0.7)
MADPO (Same-size)‡	**92.9**	58.8	15.6	41.5
MADPO (2×)	61.3 (↓31.6)	49.2 (↓9.6)	3.8 (↓11.8)	43.2 (↑1.7)
MADPO (4×)	71.8 (↓21.1)	49.8 (↓9.0)	21.9 (↑6.3)	46.9 (↑5.4)
MADPO (*Deepseek-Pro*)	74.6 (↓18.3)	**65.7** (↑6.9)	22.7 (↑7.1)	47.4 (↑5.9)
(c) MARLHF Method				
MARLHF (MAGRPO)†	**86.3**	**61.0**	**22.7**	**70.5**
MARLHF (CoLLM-CC)	23.9 (↓62.4)	46.5 (↓14.5)	7.2 (↓15.5)	37.5 (↓33.0)
MARLHF (CoLLM-DC)	26.7 (↓59.6)	38.9 (↓22.1)	4.6 (↓18.1)	14.6 (↓55.9)

Table 2: Ablation results on (a) replay buffers, (b) comparators, and (c) MARLHF methods under Dec-Dec. Pivot entries † and ‡ reported in Table 1 use the λ-decay buffer, agent-size comparator, and MAGRPO. Parenthesized values show absolute changes relative to the corresponding pivot, where ↑ denotes an increase and ↓ denotes a decrease. Best results within each ablation are reported in **bold**, best results for each method within each ablation are underlined.

Then we compare same-family comparators (*Qwen2.5* or *Qwen3*) at $1\times$, $2\times$, and $4\times$ the agent-model size, and *DeepSeek-V4-Pro* from a different model family. Table 2(b) shows that scaling the comparator does not consistently improve performance. On `TLDR`, scaling from same-size to $4\times$ reduces MARLHF from 86.3% to 66.5% and MADPO from 92.9% to 71.8%; *DeepSeek-V4-Pro* achieves 70.1% and 74.6%, respectively. This is due to the capacity gap between comparators and target agents. Despite receiving the same partial information under Dec-Dec, smaller agents may struggle to realize behaviors favored by stronger comparators. This gap also varies for different optimization methods. With *DeepSeek-V4-Pro*, MADPO improves over the same-size setting from 58.8% to 65.7% on `CoopHE` and from 15.6% to 22.7% on `BFCL`, whereas MARLHF changes from 61.0% to 63.5% and from 22.7% to 21.8%, respectively. On `Travel`, MADPO improves from 41.5% to 47.4%, compared with 70.5% to 71.2% for MARLHF. These results suggest that MADPO benefits more from this external comparator than MARLHF in these domains.

Finally, we compare MAGRPO with CoLLM-CC and CoLLM-DC, as MARL algorithms for MARLHF. Table 2(c) shows that MAGRPO outperforms both alternatives across all four domains, improving `TLDR` score by at least 59.6% and `CoopHE` pass rate, `BFCL` correct rate, and `Travel` pass by at least 14.5%, 15.5%, and 33.0%, respectively. We also observe less stable training with the actor-critic variants, as fitting critics to imperfect or evolving rewards may introduce additional approximation error and further destabilize policy optimization.

6 Conclusion

We studied preference-based optimization of cooperative LLM systems. We introduced MAPL, a general framework that iteratively updates agent policies by comparing their solutions with those generated by a comparator policy. We instantiated MAPL with MARLHF and MADPO, and evaluated them under the Dec-Dec, Cen-Cen, and Dec-Cen paradigms. Results on collaborative writing, coding, tool use, and travel planning show that MAPL improves both decentralized collaboration and centralized collaboration, and approaches the performance of MARL with well-specified rewards. Our ablations further show that a larger or centralized comparator does not necessarily facilitate training, due to the information and realizability gaps between agents and comparators.

CODE AND REPRODUCIBILITY

Our code is available at: `https://anonymous.4open.science/r/MAPL-ACPT`. We will open-source our code in the camera-ready version of the paper.

LLM USE DECLARATION

We used LLMs to correct grammatical errors, polish language, and assist in searching for related work.

REFERENCES

Stefano V Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press, 2024.

Christopher Amato. An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*, 2024.

Anthropic. Introducing the model context protocol. `https://www.anthropic.com/news/model-context-protocol`, November 2024. Accessed: 2026-07-28.

Anthropic. How we built a multi-agent research system, 2025. URL `https://www.anthropic.com/engineering/built-multi-agent-research-system`.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Paul Barde, Jakob Foerster, Derek Nowrouzezahrai, and Amy Zhang. A model-based solution to the offline multi-agent reinforcement learning coordination problem. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 141–150, 2024.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

The Viet Bui, Tien Anh Mai, and Thanh Hong Nguyen. O-MAPL: Offline multi-agent preference learning. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5811–5841. PMLR, 2025. URL `https://proceedings.mlr.press/v267/bui25a.html`.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. RLHF deciphered: A critical analysis of reinforcement learning from human feedback for llms. *ACM Computing Surveys*, 58 (2):1–37, 2025.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Association for Computational Linguistics*, pp. 7066–7085, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Weize Chen, Ziming You, Ran Li, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence. In *International Conference on Learning Representations*, volume 2025, pp. 36374–36411, 2025.

Yiqun Chen, Lingyong Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. Improving retrieval-augmented generation through multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 134495—134526, 2026.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in neural information processing systems*, volume 30, 2017.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 2, pp. 746–752, 1998.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *International Conference on Machine Learning*, volume 235, pp. 9722–9744. PMLR, 2024.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, volume 235, pp. 11733–11763. PMLR, 2024.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online AI feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zilin Wang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Sehyeok Kang, Minu Kim, Jihwan Oh, and Se-Young Yun. Dual preference learning for multi-agent reinforcement learning. *IEEE Access*, 14:113–130, 2025.

Chinmaya Kausik, Mirco Mutti, Aldo Pacchiano, and Ambuj Tewari. A theoretical framework for partially-observed reward states in rlhf. In *International Conference on Learning Representations*, volume 2025, pp. 22669–22719, 2025.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for RL. In *International Conference on Learning Representations*, 2023.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. In *Advances in neural information processing systems*, volume 36, pp. 47669–47681, 2023.

Qian Kou, Mingyang Li, Zeyang Liu, Long Qian, Zhuoran Chen, Lipeng Wan, Xingyu Chen, and Xuguang Lan. Offline multi-agent preference-based reinforcement learning with agent-aware direct preference optimization. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1181–1190, 2025.

Kimin Lee, Laura M. Smith, and Pieter Abbeel. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, volume 139, pp. 6152–6163. PMLR, 2021.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Advances in neural information processing systems*, volume 36, pp. 51991–52008, 2023.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. Marft: Multi-agent reinforcement finetuning. *arXiv preprint arXiv:2504.16129*, 2025.

Shuo Liu, Tianle Chen, Ryan Amiri, and Christopher Amato. Learning decentralized llm collaboration with multi-agent actor critic. *arXiv preprint arXiv:2601.21972*, 2026a.

Shuo Liu, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32150–32158, 2026b.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124198–124235, 2024.

João Messias, Matthijs Spaan, and Pedro Lima. Efficient offline communication policies for factored multiagent pomdps. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

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

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International conference on machine learning*, pp. 17221–17237. PMLR, 2022.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30215–30248, 2025.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *International Conference on Machine Learning*, 2025.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Noam Razin, Yong Lin, Jiarui Yao, and Sanjeev Arora. Why is your language model a poor implicit reward model? In *International Conference on Learning Representations*, 2026.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 2186–2188, 2019. Extended abstract.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, et al. Multi-turn reinforcement learning with preference human feedback. In *Advances in Neural Information Processing Systems*, 2024.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77290–77312, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Ruizhe Shi, Minhak Song, Runlong Zhou, Zihan Zhang, Maryam Fazel, and Simon S Du. Understanding the performance gap in preference learning: A dichotomy of RLHF and DPO. In *International Conference on Machine Learning*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in neural information processing systems*, volume 36, pp. 8634–8652, 2023.

Yashar Talebirad and Amirhossein Nadiri. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.

Ming Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *International Conference on Machine Learning*, pp. 330–337, San Francisco, CA, USA, 1993. Morgan Kaufmann.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the workshop on new frontiers in summarization*, pp. 59–63, 2017.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Shijie Wang, Pengfei Li, Yikun Fu, Kaifeng Liu, Fangyuan Li, Yang Liu, Xiaowei Sun, Zonglin Li, Siyao Zhao, Jian Zhao, et al. MARTI-MARS^2: Scaling multi-agent self-search via reinforcement learning for code generation. *arXiv preprint arXiv:2602.07848*, 2026.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint. In *International Conference on Machine Learning*, volume 235, pp. 54715–54754. PMLR, 2024.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107309–107336, 2026.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10299–10312, 2021.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *International Conference on Learning Representations*, 2024.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Kaiyan Zhang, Kai Tian, Runze Liu, Sihang Zeng, Xuekai Zhu, Guoli Jia, Yuchen Fan, Xingtai Lv, Yuxin Zuo, Che Jiang, Yuru wang, Jianyu Wang, Ermo Hua, Xinwei Long, Junqi Gao, Youbang Sun, Zhiyuan Ma, Ganqu Cui, Ning Ding, Biqing Qi, and Bowen Zhou. Marti: A framework for multi-agent llm systems reinforced training and inference. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138790–138813, 2026.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. Stronger-MAS: Multi-agent reinforcement learning for collaborative LLMs. In *International Conference on Learning Representations*, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

Tianchen Zhu, Yue Qiu, Haoyi Zhou, and Jianxin Li. Decoding global preferences: Temporal and cooperative dependency modeling in multi-agent preference-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17202–17210, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.