OPTIMA: Optimizing Effectiveness and Efficiency for LLM-Based Multi-Agent System

Anonymous ACL submission

Abstract

Large Language Model (LLM) based multi-agent systems (MAS) show remarkable potential in collaborative problem-solving, yet they still face critical challenges: low communication efficiency, poor scalability, and a lack of effective parameter-updating optimization methods. We present **OPTIMA**, a novel framework that addresses these issues by significantly enhancing *both* communication efficiency and task effectiveness in LLM-based MAS through training. OPTIMA employs an *iterative generate, rank, select, and train* paradigm with a reward function balancing task performance, token efficiency, and communication readability. We explore various algorithms, including Supervised Fine-Tuning, Direct Preference Optimization, and their hybrid approaches, providing insights into their effectiveness-efficiency trade-offs. We integrate Monte Carlo Tree Search-inspired techniques for DPO data generation, treating conversation turns as tree nodes to explore diverse interaction paths. Evaluated on common multi-agent tasks, including information-asymmetric question answering and complex reasoning, OPTIMA shows consistent and substantial improvements over single-agent baselines and vanilla MAS based on Llama 3 8B / 3.2 3B, achieving up to *2.8x performance gain with less than 10% tokens* on tasks requiring heavy information exchange. Moreover, OPTIMA's efficiency gains enable more effective compute utilization during inference, leading to improved inference-time scaling laws. By addressing fundamental challenges in LLM-based MAS, OPTIMA shows the potential towards scalable, efficient, and effective MAS.

1 Introduction

Large Language Models (LLMs) have emerged as powerful tools for a wide range of tasks, from natural language processing to complex reasoning (OpenAI, 2023; Reid et al., 2024; Anthropic, 2024).

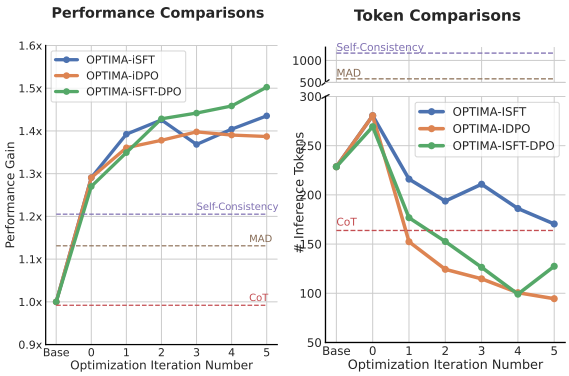


Figure 1: **Performance and efficiency of OPTIMA variants across optimization iterations. Left**: Average performance gain over iterations. OPTIMA variants consistently outperform CoT, Multi-Agent Debate (MAD), and Self-Consistency. **Right**: Average inference token numbers over iterations. All OPTIMA variants achieve better performance with substantially fewer tokens.

A promising direction in leveraging these models is the development of autonomous multi-agent systems (MAS), which aim to harness the collective intelligence of multiple LLM-based agents for collaborative problem-solving and decision-making (Liang et al., 2023; Wang et al., 2024b; Du et al., 2024; Zhuge et al., 2024). However, for LLM-based MAS to be truly effective, they must overcome two critical challenges: **(a)** achieving efficient inter-agent communication to minimize computational costs, and **(b)** optimizing the collective performance of the system as a cohesive unit.

Current LLM-based MAS face significant difficulties in meeting these challenges. The coordination and communication between agents often lack efficiency, resulting in verbose exchanges that lead to increased token usage, longer inference times, and higher computational costs (Li et al., 2024b). This inefficiency is exacerbated by the *length bias* inherent in LLMs due to alignment training (Saito et al., 2023; Dubois et al., 2024), which favors longer responses even when concise communication would suffice (Chen et al., 2024c). Moreover,

while recent work has explored training LLMs for single-agent tasks (Song et al., 2024; Xiong et al., 2024) and MAS training is well-studied in reinforcement learning (Johnson et al., 2000; Lanctot et al., 2017; Baker et al., 2020), there remains a lack of parameter-updating methods specifically designed to optimize LLM-based MAS as a unified system. Existing approaches primarily rely on simple agent profile evolution (Chen et al., 2024b) or memory evolution (Qian et al., 2024a,b; Gao et al., 2024), which fail to address the core issues of communication efficiency and collective optimization.

Can we develop a training framework that simultaneously enhances the communication efficiency and task effectiveness of LLM-based MAS? To address this question, we introduce **OPTIMA**, an effective framework designed to optimize LLM-based MAS. At the heart of OPTIMA is an iterative *generate, rank, select, and train* paradigm, incorporating a reward function that balances task performance, token efficiency, and communication readability. This approach enables the development of MAS that are not only effective and efficient but also maintain interpretable communication patterns. Based on the reward function, OPTIMA leverages a combination of techniques to induce efficient and effective communication behaviors in LLM-based agents, including Supervised Fine-Tuning (SFT) (Zelikman et al., 2022; Gülçehre et al., 2023; Aksitov et al., 2023) and Direct Preference Optimization (DPO) (Rafailov et al., 2023; Pang et al., 2024), along with their hybrid variants. Furthermore, OPTIMA introduces an integration of Monte Carlo Tree Search (MCTS)-inspired techniques for DPO data generation, conceptualizing conversation turns as tree nodes to explore diverse interaction trajectories efficiently.

Importantly, by substantially reducing the number of tokens required for inference, OPTIMA not only improves computational efficiency but also opens new possibilities for leveraging inference compute more effectively. This reduction in token usage allows for more samples within the same computational constraints, potentially leading to *better inference-time scaling laws*. As recent work has shown the importance of inference-time compute in improving model performance (Wu et al., 2024; Brown et al., 2024; Chen et al., 2024a), OPTIMA's efficiency gains could be combined with techniques like majority voting (Wang et al., 2023), leading to more effective LLM systems.

We evaluate OPTIMA on a diverse set of tasks spanning two multi-agent settings: **(a)** information exchange, including information-asymmetric question answering (Chen et al., 2024c; Liu et al., 2024), and **(b)** debate, encompassing mathematical and reasoning tasks (Du et al., 2024; Chen et al., 2024b; Wu et al., 2023). Using Llama 3 8B / 3.2 3B (Meta, 2024) as our base model, we demonstrate that OPTIMA consistently outperforms both single-agent MAS baselines, achieving up to 90% reduction in token usage and 2.8x increase in task performance.

To summarize, our main contribution is OPTIMA, a novel training framework that simultaneously optimizes *communication efficiency* and *task effectiveness*. To enhance high-quality training data generation *in multi-agent settings* for DPO, we introduce an integration of MCTS-like techniques. Our comprehensive empirical evaluation across diverse tasks demonstrates notable advancements in *both* token efficiency and task performance, while also providing insights into the learned communication patterns. Additionally, we examine the implications of OPTIMA's efficiency gains for inference-time scaling, underscoring its potential to improve the LLM systems by enabling more effective utilization of inference-compute. By addressing the dual challenges of communication efficiency and collective optimization, our work underscores the importance of developing advanced training frameworks for LLM-based MAS and highlights efficiency as a crucial metric to consider. We believe OPTIMA provides a solid foundation for future investigations into scaling and improving MAS and general LLM systems.

2 OPTIMA: Optimizing Multi-Agent LLMs via Iterative Training

2.1 Overview

OPTIMA is built upon an iterative *generate, rank, select, and train* paradigm. This approach allows for the progressive improvement of LLM-based agents in multi-agent settings, focusing on enhancing both the efficiency of inter-agent communication and the effectiveness of task completion.

Let $\mathcal{M}_{\text{base}}$ denote the base LLM, $\mathcal{D}$ the task dataset, and f the iterative training function. The iterative process can be formalized as $\mathcal{M}_{t+1} = f(\mathcal{M}_t, \mathcal{D})$, where $\mathcal{M}_t$ represents the model at iteration t. The function f encapsulates the entire process of data generation, ranking, selection and model training. For each task instance $d_i \in \mathcal{D}$, we sample a set of N conversation trajectories

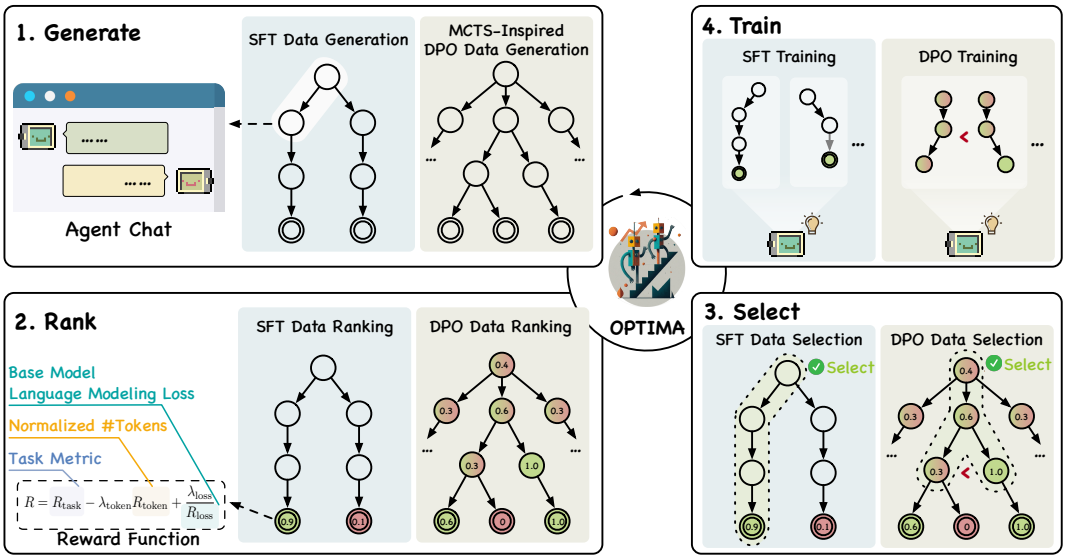


Figure 2: **Overview of the OPTIMA framework for training LLM-based MAS**. The iterative process includes four stages: *Generate, Rank, Select*, and *Train*. Note that the ranking process, while also involved in DPO data generation, is not shown in the Generate stage for simplicity.

$\{\tau_i^j\}_{j=1}^N \subset \mathcal{T}$ using the agents powered by current model $\mathcal{M}_t$. Each trajectory τ_i^j is then evaluated using a reward function $R : \mathcal{T} \to \mathbb{R}$, defined as:

$$R(\tau_i^j) = R_{\text{task}}(\tau_i^j) - \lambda_{\text{token}} R_{\text{token}}(\tau_i^j) + \lambda_{\text{loss}} \frac{1}{R_{\text{loss}}(\tau_i^j)}. \quad (1)$$

Here, $R_{\text{task}} : \mathcal{T} \to \mathbb{R}$ is the task-specific performance metric, $R_{\text{token}}(\tau_i^j) = \frac{\#\text{Tokens}(\tau_i^j)}{\max_k(\{\#\text{Tokens}(\tau_i^k)\}_k)}$ is the normalized token count, and $R_{\text{loss}}(\tau_i^j) = g(\mathcal{L}(\mathcal{M}_{\text{base}}, d_i, \tau_i^j))$ is based on the language modeling loss of the base model $\mathcal{M}_{\text{base}}$, which we detail in Appendix H.2. The positive coefficients λ_{token} and λ_{loss} are hyper-parameters . This reward function is designed to balance multiple objectives simultaneously: R_{task} ensures that the model improves on the intended task, R_{token} encourages communication efficiency by penalizing verbose exchanges, and R_{loss} regularizes language naturalness and readability by favoring trajectories that are probable under the base model. By incorporating these components, we aim to develop LLM-based MAS that are not only effective in their designated tasks but also efficient in their communication, while maintaining interpretability in their outputs, unlike the often incomprehensible communication in prior RL research (Lazaridou et al., 2017; Evtimova et al., 2018; Chaabouni et al., 2022).

Based on these rewards, we apply several data selection criteria to select a subset of high-quality sampled trajectories $\{\tau_i^*\}$ for each task instance. These selected trajectories form the training data $\mathcal{D}_i^*$ at iteration i. The model is then updated: $\mathcal{M}_{t+1} = \text{Train}(\mathcal{M}_t, \mathcal{D}_i^*)$. The Train function can be instantiated with various training algorithms, such as SFT or DPO, which we will discuss in detail in the following subsections.

Fig. 2 provides a high-level overview of OPTIMA. The specific instantiations of the generation and training processes will be detailed in the following subsections. The ranking process, consistent across all instantiations, is defined by the reward function presented in Eq. (1).

2.2 Initialization

Before starting the iterative training process, we address a critical challenge in LLM-based MAS: agents often produce responses in a similar style across conversation trajectories, even with high-temperature sampling. This homogeneity limits the exploration of diverse communication strategies, potentially hindering the optimization toward more efficient and effective interactions. Following the observation from AutoForm (Chen et al., 2024c), where LLMs can be explicitly prompted to leverage different more concise formats to communicate or reason without much compromise in performance, we introduce an initialization step that promotes diversity in agent communication.

Our approach leverages a pool of format specification prompts, $\mathcal{P} = \{p_1, p_2, ..., p_K\}$, where each p_k is a string specifying a particular response format (e.g., JSON, list, see Appendix I for concrete examples and creation process). For each task instance $d_i \in \mathcal{D}$, we generate N conversation tra-

jectories, each with a randomly selected format specification appended to the input task:

$$\tau_i^j = \mathcal{M}_{\text{base}}(d_i \oplus p_{k_j}),\ k_j \sim \text{Uniform}(1, K), \quad (2)$$

where $\oplus$ denotes string concatenation. This process yields a diverse set of trajectories $\{\tau_i^j\}_{j=1}^N$ for each d_i, varying in both content and structure.

We then evaluate these trajectories using the reward function defined in Eq. (1), for each d_i, we select the trajectory with the highest reward: $\tau_i^* = \arg\max_j R(\tau_i^j)$. Finally, we select top 70% trajectories that exceed a predefined performance threshold θ_{init}, resulting in a high-quality dataset:

$$\mathcal{D}_0^* = \text{TopK}(\{(d_i, \tau_i^*) \mid R_{\text{task}}(\tau_i^*) > \theta_{\text{init}}, \forall d_i \in \mathcal{D}\}, 70\%). \quad (3)$$

Crucially, we remove the format specification prompts from the selected trajectories, resulting in a dataset of diverse, high-quality conversations without explicit format instructions. We then fine-tune the base model $\mathcal{M}_{\text{base}}$ to obtain $\mathcal{M}_0 = \text{SFT}(\mathcal{M}_{\text{base}}, \mathcal{D}_0^*)$, which serves as the starting point for OPTIMA, able to generate diverse communication patterns without explicit format prompting. We provide pseudo-code in Appendix B for better understanding. This initialization sets the stage for more effective exploration and optimization in the subsequent iterative training process.

2.3 Instantiation 1: Iterative SFT

We introduce iterative Supervised Fine-Tuning (iSFT) as our first instantiation of OPTIMA. At each iteration t, iSFT follows the same general procedure outlined in Algorithm 1, generating a set of N conversation trajectories for each task training instance $d_i \in \mathcal{D}$ using the current model $\mathcal{M}_t^{\text{iSFT}}$. However, unlike initialization, iSFT omits the format specification pool, as $\mathcal{M}_0$ has already internalized diverse communication strategies. Unlike recent research on iterative training (Gülçehre et al., 2023; Aksitov et al., 2023), iSFT maintains a fixed reward threshold θ_{SFT} across iterations for data selection. The model is then trained with standard SFT. This process continues until a maximum number of iterations is reached. For clarity, the pseudo-code for iSFT is provided in Appendix B.

iSFT provides a straightforward yet effective approach to optimize LLM-based MAS, leveraging the diverse communication patterns established during initialization while consistently improving task performance and communication efficiency.

2.4 Instantiation 2: Iterative DPO

While iSFT provides a straightforward approach to optimizing LLM-based MAS, it may be limited by its reliance on a single *best* trajectory for each task instance. To address this, we explore iterative Direct Preference Optimization (iDPO) (Rafailov et al., 2023; Pang et al., 2024), which optimizes models using comparative preferences and has demonstrated success in LLM alignment. Applying DPO in multi-agent settings, however, poses distinct challenges, particularly in generating meaningful paired data that capture the complexities of agent interactions.

Data Generation: To overcome these challenges, we integrate MCTS with DPO data collection for high-quality paired data generation in multi-agent settings. Our MCTS-based approach conceptualizes the multi-agent conversation as a tree, where nodes represent conversational turns, and edges represent continuations. This structure allows us to explore diverse interaction trajectories systematically and select high-quality paired data for DPO training. The MCTS process begins at the root node (initial task prompt) and proceeds as follows: **(1) Expansion**: We select a node to expand based on the following criteria. We first exclude leaf nodes and the second-to-last level nodes to avoid wasting computation on low-variance expansions, then exclude nodes with content similar to previously expanded nodes, measured based on edit distance (see Appendix H.1). From the remaining nodes, we select 10 nodes with the highest rewards and sample one using the softmax distribution over their rewards. **(2) Simulation**: For each selected node, we expand 3 trajectories, simulating the conversation to completion. **(3) Backpropagation**: Once a trajectory is completed and rewarded with Eq. (1), we update the estimated rewards of all nodes in the trajectory with the average rewards from their children. **(4) Iteration**: We repeat the above process 8 times, resulting in 24 trajectories. More iterations could potentially lead to more diverse and better-quality data.

Paired Data Construction: To generate high-quality paired data for DPO training, we traverse each MCTS tree and identify node pairs (n_i, n_j) that satisfy three conditions: (1) shared ancestry, (2) the higher estimated reward of n_i and n_j exceeds the threshold $\theta_{\text{dpo-filter}}$, and (3) their reward difference exceeds the threshold $\theta_{\text{dpo-diff}}$. We sort these pairs by the higher estimated reward, and se-

lect the top 50% pairs as part of the final training set. We construct DPO training instances by using the common conversation history as the prompt, with n_i and n_j serving as the chosen and rejected responses according to their estimated rewards.

The iDPO process then proceeds iteratively, alternating between MCTS-based data generation and model updates using DPO. The pseudo-code for our iDPO process is presented in Appendix B.

2.5 Instantiation 3: Hybrid Iterative Training

Building upon the strengths of both iSFT and iDPO, we investigate a hybrid approach that interleaves SFT and DPO in the iterative training process, termed as iSFT-DPO. This hybrid method aims to leverage the simplicity and directness of SFT in capturing high-quality trajectories, while also benefiting from the nuanced comparative learning facilitated by DPO. By alternating between these two training paradigms, we hypothesize that the model can more effectively balance the exploration of diverse communication strategies with the exploitation of known effective patterns.

In practice, we implement this hybrid approach by performing one iteration of iSFT followed by one iteration of iDPO, and repeating this cycle throughout the training process. This interleaving allows the model to first consolidate learning from the best observed trajectories through SFT, and then refine its understanding through the comparative preferences provided by DPO.

3 Experiments

Datasets. We evaluate OPTIMA in two settings: information exchange (IE) and debate. For IE, we use HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2WMHQA) (Ho et al., 2020), TriviaQA (Joshi et al., 2017), and CBT (Hill et al., 2016). For multi-hop datasets (HotpotQA, 2WMHQA), we split relevant contexts between two agents, ensuring the answer can only be deduced from information exchange. For TriviaQA and CBT, contexts are randomly assigned, challenging agents to communicate and identify the relevant information. The debate setting employs GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), ARC's challenge set (ARC-C) (Bhakthavatsalam et al., 2021) and MMLU (Hendrycks et al., 2021a), with one agent as solver and another as critic (Chen et al., 2024b). We use 0-shot for all benchmarks.

Metrics. We report F1 score between generated answers and labels for IE tasks. For debate tasks, we employ exact match accuracy (GSM8k, ARC-C, MMLU) or Sympy-based (Meurer et al., 2017) equivalence checking (MATH), following Lewkowycz et al. (2022). Conversations conclude when agents both mark the same answer with specified special tokens or reach a turn limit.

Baselines. We compare against single-agent approaches: Chain-of-Thought (CoT) (Wei et al., 2022) and Self-Consistency (SC) with majority voting (Wang et al., 2023) on $n = 8$ samples. For IE tasks, direct majority voting is impractical due to free-form responses. Instead, we compute pairwise F1 scores, group answers with scores above 0.9, and report the average F1 score of the largest group against the label. In multi-agent settings, we compare against Multi-Agent Debate (MAD) (Du et al., 2024) and AutoForm (Chen et al., 2024c). MAD uses natural language for inter-agent communication, while AutoForm employs concise, non-natural-language formats for better performance-cost efficiency.

Training Setups. We use Llama 3 8B / 3.2 3B (Meta, 2024) as our base model, focusing on two-agent scenarios without external tools to isolate core multi-agent communication and collaboration. A single model is trained for both agents, with separate model training left for future work. Iterative training completes within 12 hours on 8 A100 GPUs for most tasks, except MATH, which takes around 24 hours. More details are in Appendices H and I.

3.1 Benchmark Results

Table 1 showcases OPTIMA's performance across diverse tasks, revealing consistent improvements in effectiveness and efficiency. For IE tasks, OPTIMA variants excel, particularly in multi-hop reasoning like HotpotQA and 2WMHQA. iSFT-DPO achieves the best performance while significantly reducing token usage compared to SC. Notably, on 2WMHQA, iSFT-DPO improves F1 by **38.3%** (2.8x) while using only **10%** of MAD's tokens. This efficiency extends to other IE tasks, where OPTIMA variants maintain high performance with drastically lower token counts.

In debate tasks, OPTIMA's benefits are nuanced but evident. It achieves better performance and efficiency on ARC-C and MMLU, while on MATH and GSM8k, OPTIMA variants show comparable or slightly lower performance than SC, but with much higher token efficiency. We attribute this to task

	Information Exchange								Debate							
	HotpotQA		2WMH QA		TriviaQA		CBT		MATH		GSM8k		ARC-C		MMLU	
Method	**F1**	**#Tok**	**F1**	**#Tok**	**F1**	**#Tok**	**F1**	**#Tok**	**Acc**	**#Tok**	**Acc**	**#Tok**	**Acc**	**#Tok**	**Acc**	**#Tok**
CoT	25.6	123.7	20.5	139.8	59.8	110.3	43.4	135.3	23.9	329.8	71.5	230.9	65.2	138.9	46.0	132.2
SC ($n = 8$)	33.8	996.3	28.7	1052.8	70.0	891.4	52.9	1067.7	**35.7**	2600.9	80.3	1828.7	75.6	1116.7	54.0	1056.1
MAD	28.4	570.9	25.9	543.7	71.0	408.6	53.8	493.0	29.8	1517.6	72.5	514.7	71.4	478.0	51.5	516.7
AutoForm	28.2	97.7	24.7	117.7	60.9	74.0	35.0	64.8	26.1	644.3	71.0	410.5	60.2	221.2	43.8	198.5
OPTIMA-iSFT	54.5	67.6	72.4	61.2	71.9	51.5	**71.8**	38.5	30.1	830.3	79.5	311.5	74.1	92.2	56.8	123.8
OPTIMA-iDPO	52.5	**45.7**	66.1	**35.9**	69.3	69.2	66.7	**37.2**	30.4	**272.8**	78.5	270.1	74.5	97.8	59.6	61.6
OPTIMA-iSFT-DPO	**55.6**	63.3	**74.2**	54.9	**77.1**	**32.5**	70.1	38.9	29.3	488.1	**80.4**	**246.5**	**77.1**	**88.0**	**60.2**	**56.7**
OPTIMA-iSFT SC	54.8	806.2	72.6	245.6	73.7	413.8	72.2	847.4	32.4	2432.9	83.1	1750.7	77.2	1148.7	60.2	874.5
OPTIMA-iDPO SC	52.8	412.8	67.2	1056.2	71.8	702.8	66.8	520.6	36.9	2743.1	84.4	1750.8	77.0	1091.2	59.9	1050.4
OPTIMA-iSFT-DPO SC	57.4	957.9	76.7	1096.0	77.5	494.1	71.8	417.8	34.8	2788.5	84.0	1748.7	78.8	1036.1	61.2	1026.7

Table 1: **Performance and inference token number comparison across information exchange and debate tasks.** Best results are indicated in **bold**, and second-best results are underlined for all rows except the last three. The last three rows display self-consistency results for OPTIMA variants, with the best results highlighted in green. OPTIMA variants consistently outperform baselines in task performance and/or token efficiency.

	2WMH QA		Trivia QA		GSM8k	
Method	**F1**	**#Tok**	**F1**	**#Tok**	**Acc**	**#Tok**
MAD	25.9	543.7	71.0	408.9	72.5	514.7
AutoForm	24.7	117.7	60.9	74.0	71.0	410.5
iSFT	**56.5**	79.6	70.0	90.2	74.6	293.7
iDPO	51.6	84.3	68.0	**41.1**	**77.9**	**185.7**
iSFT-DPO	54.5	**70.4**	**72.0**	67.8	74.2	363.1

Table 2: **Transfer performance of OPTIMA.** We transfer OPTIMA from Hotpot QA to 2WMH QA and Trivia QA, and from MATH to GSM8k, with MAD and AutoForm on each target task as baselines.

difficulty and limited training data. Nevertheless, Section 3.2 will show OPTIMA models trained on MATH transfer effectively to GSM8k, achieving near-equivalent performance with high efficiency. Additionally, Section 3.3 will demonstrate that applying SC to OPTIMA variants trained on MATH or GSM8k greatly improves inference scaling laws on GSM8k compared to CoT SC.

Among OPTIMA variants, iSFT often prioritizes performance at the cost of efficiency, while iDPO achieves remarkable token reductions, sometimes with performance trade-offs. iSFT-DPO strikes a robust balance, frequently delivering top-tier performance with satisfying efficiency. Results on Llama 3.2 3B in Appendix F further validate OPTIMA's robustness.

3.2 How Well Does OPTIMA Generalize?

To assess OPTIMA's ability to generalize, we conducted transfer learning experiments across different task domains. We transferred models trained on HotpotQA to TriviaQA and 2WMHQA, as well as transferring from MATH to GSM8k. While these datasets share broad categories (question-answering and mathematical reasoning, respectively), they present different challenges in terms of complexity and required skills. The results, presented in Table 2, demonstrate OPTIMA's robust transferability across these diverse tasks. In the question-answering domain, all OPTIMA variants significantly outperform baseline multi-agent methods on both OOD datasets. On 2WMHQA, the transferred iSFT more than doubles MAD's F1 score while using only 14.6% of the tokens. Similar trends are observed in TriviaQA. When transferring from MATH to GSM8k, OPTIMA variants, particular iDPO, not only outperform the baselines on GSM8k but also achieve results comparable to models directly trained on GSM8k with even higher token efficiency (refer to Table 1 for comparison).

These results underscore OPTIMA's potential for developing adaptable MAS, demonstrating that OPTIMA-trained models learn transferable skills for efficient information exchange and collaborative reasoning. However, transferring to more distant domains remains challenging, e.g., we find it hard to transfer from from MATH to ARC-C. We believe it is a promising area for future research to explore if scaling OPTIMA to more generalized multi-task training could enhance the generalization.

3.3 Can OPTIMA Improve Inference Scaling?

Recent research emphasizes inference-time scaling, which describes how model performance improves with increased compute during inference, typically by generating multiple samples per problem (Brown et al., 2024; Wu et al., 2024). Unlike training scaling laws, which focus on model size, dataset size, and performance, inference-time scaling explore the trade-off between compute budget and task accuracy, offering a promising way to en-

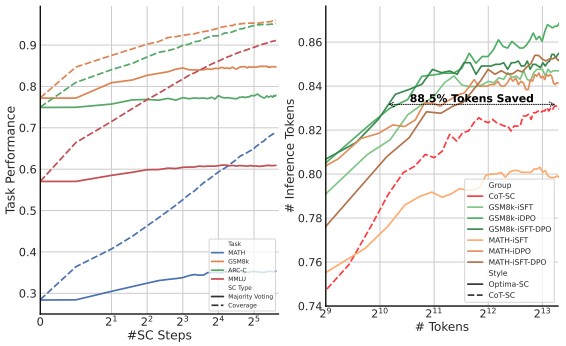


Figure 3: **OPTIMA's impact on inference scaling laws. Left** Relationship between OPTIMA variants' self-consistency steps and performance on debate tasks. Solid lines represent majority voting accuracy, while dashed lines show coverage. **Right** Performance of various models on GSM8k as a function of token usage, demonstrating OPTIMA's efficiency gains.

hance model capabilities without further training.

Fig. 3 illustrates OPTIMA's impact on inference-time scaling. The left panel shows the relationship between SC steps and performance on multi-agent debate tasks. While majority voting accuracy plateaus after a certain number of steps, coverage (the percentage of problems answered correctly at least once) improves logarithmically with increased sampling. This aligns with recent studies (Wu et al., 2024; Chen et al., 2024a), suggesting advanced answer selection techniques could further boost OPTIMA's performance. Additional scaling law figures for all OPTIMA variants and tasks are in Appendix A, where similar trends are observed.

The right panel demonstrates OPTIMA's efficiency in improving inference scaling laws on GSM8k. OPTIMA variants, including those transferred from MATH, consistently outperform CoT SC, except for MATH-trained iSFT. Notably, GSM8k-trained iDPO matches CoT-SC performance with 88.5% fewer tokens, effectively "*shifting the curve left*". This reduction in token usage translates to significant computational savings without sacrificing accuracy. MATH-trained OPTIMA variants, except iSFT, also deliver better scaling laws on GSM8k than CoT SC, highlighting OPTIMA's cross-task generalization.

These results underscore OPTIMA's potential to reshape inference-time scaling for MAS and general LLM systems. By enabling more efficient use of compute budgets, OPTIMA achieves better performance at lower costs or higher performance at the same cost. This efficiency opens possibilities for integrating advanced inference techniques like weighted voting or tree-search (Wu et al., 2024), potentially leading to further performance gains.

	2WMH QA		ARC-C	
Setting	**F1**	**#Tok**	**Acc**	**#Tok**
iSFT	**72.4**	61.2	74.1	92.2
w/o #Tokens	**72.4**$_{(0.0)}$	290.3$_{(4.8x)}$	**74.2**$_{(+0.1)}$	579.6$_{(6.3x)}$
w/o Loss	69.7$_{(-2.7)}$	**45.4**$_{(0.7x)}$	72.6$_{(-1.5)}$	**69.7**$_{(0.8x)}$
iDPO	66.1	**35.9**	74.5	97.8
w/o #Tokens	**72.9**$_{(+6.8)}$	183.3$_{(5.1x)}$	**75.5**$_{(+1.0)}$	266.0$_{(2.7x)}$
w/o Loss	63.0$_{(-3.1)}$	54.6$_{(1.5x)}$	74.4$_{(-0.1)}$	**81.2**$_{(0.8x)}$
iSFT-DPO	**74.2**	54.9	**77.1**	88.0
w/o #Tokens	63.5$_{(-10.7)}$	219.7$_{(4.0x)}$	76.9$_{(-0.2)}$	354.8$_{(4.0x)}$
w/o Loss	66.7$_{(-7.5)}$	**38.1**$_{(0.7x)}$	76.3$_{(-0.8)}$	**63.4**$_{(0.7x)}$

Table 3: Ablation study on reward components for OPTIMA variants on two representative tasks.

3.4 How Does OPTIMA Evolve Performance?

To understand the impact of reward function components in our reward function, we conducted an ablation study on 2WMHQA (IE) and ARC-C (debate). We removed either token count regularization (#Tokens) or LM loss (Loss) to address: **(1)** *How does token count regularization affect efficiency-performance trade-offs?* **(2)** *What role does LM loss play in maintaining communication quality?* Our findings highlight the importance of each component in balancing performance, efficiency, and language quality.

Table 3 presents the results of our ablation study. Removing the token count leads to a substantial increase in the number of generated tokens across settings, with a particularly pronounced effect in the debate task. While this increased verbosity occasionally results in marginal performance improvements, it comes at a significant computational cost. Conversely, eliminating the LM loss results in a decrease in token usage, often producing the most concise outputs among all variants. Examples comparing communication with and without LM loss can be found in Appendix C. Without LM loss, the model often generates overly concise messages containing insufficient information and is prone to hallucination, potentially explaining the inferior performance. Overall, OPTIMA's reward function achieves the balance among task effectiveness, token efficiency and dialogue quality, enabling effective and efficient multi-agent collaboration.

3.5 How Agent Communication Evolves over Optimization Iterations?

Fig. 1 shows the performance gains and token efficiency of OPTIMA variants across optimization iterations, revealing a two-phase pattern. In the initial phase (iterations 0-1), all variants show significant

performance improvements alongside increased token usage, indicating OPTIMA prioritizes effectiveness by enabling agents to develop sophisticated strategies through expanded communication. In later iterations, OPTIMA refines these strategies for efficiency without sacrificing performance, with token usage decreasing gradually while performance continues to improve.

Concrete examples of OPTIMA's impact on agent communication are provided in Appendix D (iSFT on an information exchange task) and Appendix E (debate task). The base model tends to produce verbose, repetitive exchanges, while OPTIMA-trained models exhibit more concise and task-oriented communication.

3.6 Can OPTIMA Scale with More Agents?

While the previous experiments highlight OPTIMA's effectiveness in two-agent scenarios, which is a controlled setting that circumvents issues such as communication order and effectively validates the framework, we also evaluate its scalability in three-agent settings for IE and debate tasks. The results, detailed in Appendix G, demonstrate that OPTIMA continues to enhance both effectiveness and efficiency.

4 Related Work

LLM-Based MAS. LLM-powered multi-agent systems have demonstrated success in collaborative problem-solving through approaches like multi-agent debate (Liang et al., 2023; Du et al., 2024). Subsequent work explores role-playing for reasoning (Wang et al., 2024b; Chen et al., 2024b), software development (Qian et al., 2024c; Hong et al., 2024), and embodied interactions (Zhang et al., 2024; Mandi et al., 2024), with scale and diversity improving performance (Wang et al., 2024a; Li et al., 2024a). However, efficiency challenges emerge as systems grow (Chen et al., 2024c; Qian et al., 2024d), with existing methods focusing on memory updates rather than comprehensive training (Qian et al., 2024a). Our framework addresses this gap through joint optimization of communication efficiency and task effectiveness.

Iterative Refinement of LLMs. Continual improvement in LLMs has led to various iterative refinement paradigms. Self-reflection mechanisms like Reflexion (Shinn et al., 2023) and self-refine (Madaan et al., 2023) show promise but are limited by LLMs' self-correction abilities (Huang et al., 2024; Olausson et al., 2024; Kamoi et al., 2024). More robust approaches, such as ReST (Gülçehre et al., 2023), ReST^{EM} (Singh et al., 2024), and STaR (Zelikman et al., 2022), fine-tune models on self-generated high-quality reasoning paths. Pang et al. (2024) further integrate incorrect paths and train models with DPO. These methods have been extended to complex tasks (Aksitov et al., 2023), but iterative refinement in LLM-based MAS remains underexplored, as does the trade-off between effectiveness and efficiency. Our work addresses this gap by introducing the first effective training framework for iterative optimization in MAS contexts and systematically shedding light on the trade-offs between effectiveness and efficiency.

Inference-Time Scaling and Token Efficiency. Compute scaling has enhanced LLM capabilities, with approaches like majority voting and reward-guided tree search improving performance on reasoning tasks (Chen et al., 2024a; Wu et al., 2024; Brown et al., 2024; Saad-Falcon et al., 2024). However, these methods increase computational demands, highlighting the need for token efficiency. Recent work achieves efficiency through latent space reasoning via step distillation (Deng et al., 2023, 2024; Hao et al., 2024; Cheng and Durme, 2024), but at the cost of comprehensibility. Our framework advances this by (1) demonstrating iterative training framework that improves both token efficiency and task effectiveness in MAS context, and (2) showing that enhanced efficiency enables more sampling within fixed compute budgets, leading to better inference-time scaling.

5 Conclusion

We introduce OPTIMA, a novel framework for training LLM-based MAS that significantly enhances communication efficiency and task performance. Experiments show OPTIMA's consistent superiority over single-agent and multi-agent baselines. We introduce key innovations such as iterative training, a balanced reward function, and MCTS-inspired data generation. Crucially, OPTIMA effectively improves inference-time scaling and transfers effectively to OOD tasks, underscoring the importance of efficient communication in MAS and LLM systems. While OPTIMA marks a major step forward in multi-agent LLM training, further exploration into its scalability to larger models and more complex scenarios is a promising direction for future research.

Limitations

While OPTIMA demonstrates significant improvements in communication efficiency and task effectiveness for LLM-based multi-agent systems, our study has several limitations. **First**, our experiments primarily focus on two-agent scenarios with a shared model architecture, leaving open questions about scaling to larger teams (e.g., 5-10 agents) and heterogeneous agent configurations. Although preliminary results with three agents show promising trends (Section 3.6), the dynamics of larger groups may introduce new challenges in coordination efficiency that require further investigation. **Second**, while we demonstrate cross-task generalization within similar domains (e.g., MATH to GSM8k), transferring OPTIMA-trained models to substantially different application areas (e.g., from QA to math or coding) remains unexplored. **Finally**, while we evaluate on standard benchmarks, real-world deployment scenarios may involve additional constraints that our framework does not explicitly address. These limitations highlight valuable directions for future research rather than fundamental flaws, as OPTIMA's core contributions, iterative optimization with efficiency-aware rewards and MCTS-inspired data generation, provide a flexible foundation adaptable to these extensions.

References

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, Manzil Zaheer, Felix X. Yu, and Sanjiv Kumar. 2023. Rest meets react: Self-improvement for multi-step reasoning LLM agent. *CoRR*, abs/2312.10003.

Anthropic. 2024. Claude 3.5 sonnet.

Bowen Baker, Ingmar Kanitscheider, Todor M. Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2020. Emergent tool use from multi-agent autocurricula. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315.

Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. 2022. Emergent communication at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024a. Are more LLM calls all you need? towards scaling laws of compound inference systems. *CoRR*, abs/2403.02419.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Weize Chen, Chenfei Yuan, Jiarui Yuan, Yusheng Su, Chen Qian, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2024c. Beyond natural language: Llms leveraging alternative formats for enhanced reasoning and communication. *CoRR*, abs/2402.18439.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *CoRR*, abs/2412.13171.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Yuntian Deng, Yejin Choi, and Stuart M. Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *CoRR*, abs/2405.14838.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart M. Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. *CoRR*, abs/2311.01460.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. 2018. Emergent communication in a multi-modal, multi-step referential game. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Shen Gao, Hao Li, Zhengliang Shi, Chengrui Huang, Quan Tu, Zhiliang Tian, Minlie Huang, and Shuo Shang. 2024. 360{\deg}rea: Towards A reusable experience accumulation with 360{\deg} assessment for multi-agent system. *CoRR*, abs/2404.05569.

Çaglar Gülçehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling. *CoRR*, abs/2308.08998.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *CoRR*, abs/2412.06769.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jeffrey D. Johnson, Jinghong Li, and Zengshi Chen. 2000. Reinforcement learning: An introduction: R.S. sutton, A.G. barto, MIT press, cambridge, MA 1998, 322 pp. ISBN 0-262-19398-1. *Neurocomputing*, 35(1-4):205–206.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can llms actually correct their own mistakes? A critical survey of self-correction of llms. *CoRR*, abs/2406.01297.

Marc Lanctot, Vinícius Flores Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4190–4203.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024a. More agents is all you need. *CoRR*, abs/2402.05120.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024b. Improving multi-agent debate with sparse communication topology. *CoRR*, abs/2406.11776.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *CoRR*, abs/2305.19118.

Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. 2024. Autonomous agents for collaborative task under information asymmetry. *CoRR*, abs/2406.14928.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhao Mandi, Shreeya Jain, and Shuran Song. 2024. Roco: Dialectic multi-robot collaboration with large language models. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 286–299. IEEE.

Meta. 2024. Llama 3 model card.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondrej Certík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason Keith Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Stepán Roucka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony M. Scopatz. 2017. Sympy: symbolic computing in python. *PeerJ Comput. Sci.*, 3:e103.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2024. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *CoRR*, abs/2404.19733.

Chen Qian, Yufan Dang, Jiahao Li, Wei Liu, Zihao Xie, Yifei Wang, Weize Chen, Cheng Yang, Xin Cong, Xiaoyin Che, Zhiyuan Liu, and Maosong Sun. 2024a. Experiential co-learning of software-developing agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5628–5640. Association for Computational Linguistics.

Chen Qian, Jiahao Li, Yufan Dang, Wei Liu, Yifei Wang, Zihao Xie, Weize Chen, Cheng Yang, Yingli Zhang, Zhiyuan Liu, and Maosong Sun. 2024b. Iterative experience refinement of software-developing agents. *CoRR*, abs/2405.04219.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024c. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15174–15186. Association for Computational Linguistics.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2024d. Scaling large-language-model-based multi-agent collaboration. *CoRR*, abs/2406.07155.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Guha, Estefany Kelly Buchanan, Mayee F. Chen, Neel Guha, Christopher Ré, and Azalia Mirhoseini. 2024. Archon: An architecture search framework for inference-time techniques. *CoRR*, abs/2409.15254.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *CoRR*, abs/2310.10076.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2024. Beyond human data: Scaling self-training for problem-solving with language models. *Trans. Mach. Learn. Res.*, 2024.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for LLM agents. *CoRR*, abs/2403.02502.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024a. Mixture-of-agents enhances large language model capabilities. *CoRR*, abs/2406.04692.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024b. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 257–279. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. An empirical analysis of compute-optimal inference for problem-solving with language models. *CoRR*, abs/2408.00724.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch every step! LLM agent learning via iterative step-level process refinement. *CoRR*, abs/2406.11176.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

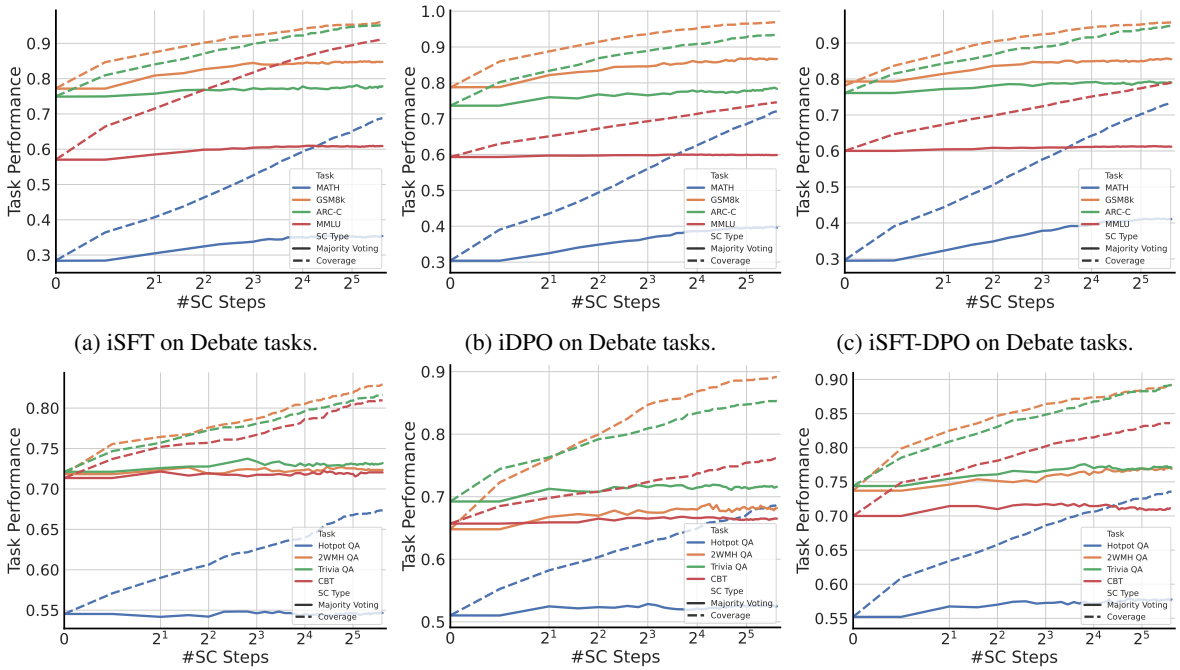
Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. 2024. Building cooperative embodied agents modularly with large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

A Inference Scaling Laws on Information Exchange Tasks

This section extends our analysis of inference scaling laws to information exchange (IE) tasks, complementing the debate task results presented in the main text (Section 3.3). Fig. 4 provides a comprehensive view of how OPTIMA variants perform across both task types as the number of SC steps increases.

For debate tasks (Fig. 4a-c), we observe consistent trends across all OPTIMA variants. The coverage exhibits a clear log-linear relationship with the number of SC steps. This trend is particularly



(a) iSFT on Debate tasks. (b) iDPO on Debate tasks. (c) iSFT-DPO on Debate tasks.

(d) iSFT on IE tasks. (e) iDPO on IE tasks. (f) iSFT-DPO on IE tasks.

Figure 4: **Inference scaling laws for OPTIMA variants on debate and information exchange (IE) tasks. (a-c)** show results for iSFT, iDPO, and iSFT-DPO on debate tasks, respectively. **(d-f)** present corresponding results for information exchange tasks. Solid lines represent majority voting accuracy, while dashed lines show coverage.

pronounced for the MATH task, where the potential for improvement through increased sampling is most evident. Majority voting accuracy tends to plateau earlier, suggesting that more sophisticated answer selection techniques might be necessary to fully leverage the diversity of generated responses.

In the case of information exchange tasks (Figures 4d-f), we note similar log-linear scaling in coverage[1] across all OPTIMA variants. However, the improvement in majority voting accuracy for IE tasks is less pronounced compared to debate tasks. This discrepancy may be attributed to the specific majority voting variant we designed for F1 scores (detailed in Section 3), which might not be optimal for capturing the nuances of partial correctness in these tasks.

These results, while highlighting some task-specific differences, collectively reinforce the potential of OPTIMA-trained models to benefit from increased inference compute. The consistent log-linear scaling in coverage across all tasks and variants indicates that there is substantial room for performance improvement through more advanced answer selection strategies or increased sampling.

[1]In IE tasks, we define coverage as the average of the highest F1 scores achieved across all generated answers for each instance.

B Additional Pseudo-Codes for OPTIMA Variants

To elucidate the implementation of various OPTIMA variants, we present algorithmic representations of several critical processes intrinsic to these variants. Specifically, we delineate the pseudo-code for **(1)** the initialization dataset collection process, as elucidated in Section 2.2 and illustrated in Algorithm 1; **(2)** the iterative supervised fine-tuning process introduced in Section 2.3 and shown in Algorithm 2; **(3)** the iteratiove DPO process as detailed in Section 2.4 and illustrated in Algorithm 3; **(4)** the Monte Carlo Tree Search-based data generation process employed in iDPO (Section 2.4), as depicted in Algorithm 5; and **(5)** the procedure for node selection during the expansion phase of MCTS, as outlined in Algorithm 4. These algorithmic representations serve to provide a comprehensive and rigorous exposition of the methodological framework underlying the OPTIMA variants.

C Case Study on Reward Components Ablation

In this section, we present a case study from the loss ablation analysis in the **iSFT-DPO** setting. In the 2WikiMultiHop QA task, we observe that with-

Algorithm 1 Initialization for Diverse Agent Communication

Input: Initial model $\mathcal{M}_0$, dataset $\mathcal{D}$, format pool $\mathcal{F}$, sample size N, reward threshold θ_{init}
Output: Initialized model $\mathcal{M}_{\text{init}}$

$\mathcal{D}^*_{\text{init}} \leftarrow \emptyset$ ▷ Initialize dataset for high-quality diverse trajectories
for each $d_i \in \mathcal{D}$ **do**
  for $j = 1$ to N **do**
    $k_j \sim \text{Uniform}(1, |\mathcal{F}|)$ ▷ Randomly select a format specification
    $\tau_i^j \leftarrow \text{AgentChat}(\mathcal{M}_0, d_i \oplus f_{k_j})$ ▷ Generate trajectory with format prompt
  end for
  $\tau_i^* \leftarrow \arg\max_j R(\tau_i^j)$ ▷ Select best trajectory
  if $R(\tau_i^*) > \theta_{\text{init}}$ **then** ▷ Check if trajectory meets quality threshold
    $\mathcal{D}^*_{\text{init}} \leftarrow \mathcal{D}^*_{\text{init}} \cup \{(d_i, \tau_i^*)\}$ ▷ Add to dataset, without format prompt
  end if
end for
$\mathcal{D}^*_{\text{init}} \leftarrow \text{TopK}(\mathcal{D}^*_{\text{init}}, 0.7|\mathcal{D}^*_{\text{init}}|)$ ▷ Retain top 70% trajectories
$\mathcal{M}_{\text{init}} \leftarrow \text{SFT}(\mathcal{M}_0, \mathcal{D}^*_{\text{init}})$ ▷ Fine-tune initial model on diverse dataset
return $\mathcal{M}_{\text{init}}$

out the constraint of the loss function, agents may generate outputs that are unreadable, contain incorrect information, and fail to communicate in a well-structured format, as demonstrated in Table 4. In the ARC task, we find that without the loss constraint, Alice tends to use fewer tokens in the reasoning process, making it harder for Bob to identify and correct errors in the reasoning, as shown in Table 5.

D Case Study on Information Exchange Task

In this section, we present a case study from iSFT on an information exchange task, with the evolution of agent communication detailed in Fig. 5.

The base model exhibits unfocused and repetitive exchanges, failing to efficiently address the task at hand. At iteration 0, while more structured, the exchange is verbose and includes unnecessary metadata. By iteration 2, we observe a marked shift towards concise, task-oriented communication, with agents adopting a streamlined format that efficiently conveys key information. The final iteration demonstrates further refinement, with agents maintaining the efficient structure while eliminating any residual verbosity. This progression aligns with our quantitative findings, showcasing OPTIMA's ability to form communication patterns that are both highly effective and remarkably efficient.

Algorithm 2 Iterative Supervised Fine-Tuning

Input: Initialized model $\mathcal{M}_{\text{init}}$, dataset $\mathcal{D}$, sample size N, reward threshold θ_{sft}, max iterations T
Output: Optimized model $\mathcal{M}_T$

$\mathcal{M}_0 \leftarrow \text{Initialize}(\mathcal{M}_{\text{init}}, \mathcal{D})$ ▷ Algorithm 1
for $t = 0$ to $T - 1$ **do**
  $\mathcal{D}^*_t \leftarrow \emptyset$
  for each $d_i \in \mathcal{D}$ **do**
    $\{\tau_i^j\}_{j=1}^N \leftarrow \text{AgentChat}(\mathcal{M}_t, d_i)$ ▷ Generate N trajectories
    $\tau_i^* \leftarrow \arg\max_j R(\tau_i^j)$ ▷ Select best trajectory
    if $R(\tau_i^*) > \theta_{\text{sft}}$ **then**
      $\mathcal{D}^*_t \leftarrow \mathcal{D}^*_t \cup \{(d_i, \tau_i^*)\}$
    end if
  end for
  $\mathcal{D}^*_t \leftarrow \text{TopK}(\mathcal{D}^*_t, 0.7|\mathcal{D}^*_t|)$ ▷ Retain top 70% trajectories
  $\mathcal{M}_{t+1} \leftarrow \text{SFT}(\mathcal{M}_t, \mathcal{D}^*_t)$
end for
return $\mathcal{M}_T$

E Case Study on Debate Task

In Appendix D, we presented an example from 2WMH QA, illustrating OPTIMA's impact on an information exchange task. Here, we provide a complementary case study from a debate task to demonstrate OPTIMA's effectiveness across different multi-agent settings. Fig. 6 showcases the evolution of agent communication in a debate task across iterations 0, 2, and 4 of OPTIMA training. The task involves discussing the environmental impact of fertilizer runoff on ocean bays.

At iteration 0, agents engage in a structured but verbose exchange. By iteration 2, the communication becomes more concise, with agents summarizing key steps without explicitly restating each link. At iteration 4, we observe further refinement in communication efficiency, with agents expressing the core concept in just three exchanges, omitting intermediate steps that can be inferred.

This progression aligns with our observations

Question: Which film was released more recently, Last Of The Long-Haired Boys or Age Of The Dragons?
Trained w/o LM Loss **Alice**: 1968 <A>Age Of The Dragons</A> **Bob**: 2011 <A>Age Of The Dragons</A>
Trained w/ LM Loss **Alice**: [["Film", "Last Of The Long-Haired Boys"], ["Release", "1968"]] **Bob**: [["Film", "Age Of The Dragons"], ["Release", "2011"]] **Alice**: <A>Age Of The Dragons</A> **Bob**: <A>Age Of The Dragons</A>

Table 4: Loss ablation on 2MultiWikiHop QA

Algorithm 3 Iterative Direct Preference Optimization

Input: Initial model $\mathcal{M}_{\text{init}}$, dataset $\mathcal{D}$, max iterations T
Output: Optimized model $\mathcal{M}_T$

$\mathcal{M}_0 \leftarrow$ Initialize($\mathcal{M}_{\text{init}}, \mathcal{D}$) ▷ Algorithm 1
for $t = 0$ to $T - 1$ **do**
  $\mathcal{D}_t^{\text{DPO}} \leftarrow \emptyset$
  for each $d_i \in \mathcal{D}$ **do**
    $\mathcal{D}_i^{\text{DPO}} \leftarrow$ MCTSDataGeneration($\mathcal{M}_t, d_i$) ▷ Algorithm 5
    $\mathcal{D}_t^{\text{DPO}} \leftarrow \mathcal{D}_t^{\text{DPO}} \cup \mathcal{D}_i^{\text{DPO}}$
  end for
  $\mathcal{M}_{t+1} \leftarrow$ DPO($\mathcal{M}_t, \mathcal{D}_t^{\text{DPO}}$)
end for
return $\mathcal{M}_T$

Algorithm 4 SelectNodeToExpand Function

Input: Tree $\mathcal{T}$, previously expanded nodes $\mathcal{N}_{\text{prev}}$, edit distance threshold ϵ, top-k k
Output: Selected node for expansion

$\mathcal{N}_{\text{eligible}} \leftarrow \{\text{n} \in \mathcal{T} \mid \text{n is not leaf and not second-to-last level}\}$
$\mathcal{N}_{\text{filtered}} \leftarrow \emptyset$
for $\text{n} \in \mathcal{N}_{\text{eligible}}$ **do**
  if $\min_{\text{n}_{\text{prev}} \in \mathcal{N}_{\text{prev}}} \text{EditDistance}(\text{n}, \text{n}_{\text{prev}}) > \epsilon$ **then**
    $\mathcal{N}_{\text{filtered}} \leftarrow \mathcal{N}_{\text{filtered}} \cup \{\text{n}\}$
  end if
end for
$\mathcal{N}_{\text{top-k}} \leftarrow \text{TopK}(\mathcal{N}_{\text{filtered}}, k, \text{key} = R(\text{n}))$
$\text{n}_{\text{selected}} \sim \text{Softmax}(\{R(\text{n}) \mid \text{n} \in \mathcal{N}_{\text{top-k}}\})$
return $\text{n}_{\text{selected}}$

in the main text, further supporting OPTIMA's capability to optimize agent communication across diverse task types. These improvements in communication dynamics contribute to both the increased task performance and reduced token consumption observed in our quantitative results, underscoring OPTIMA's versatility in training MAS to communicate effectively and efficiently.

F Results on Llama 3.2 3B

As illustrated in Section 3.1, to verify OPTIMA's ability of generalizing to different base models, we conduct experiment based on Llama 3.2 3B. The results are presented in Table 6. From the results, we can see that OPTIMA is still able to significantly improve both efficiency and performance for the model with smaller parameter sizes.

G Results on Scenarios with More Agents

Table 7 presents the results on three-agent scenarios. We select one task from both the IE task and the debate task for experimentation. It is important to note that in the debate task, we no longer designate a specific agent as the solver and another as the critic, which differs from the two-agent scenarios.

Question: Marshall researched some of the changes on Earth that happen over a long time. Which event most likely causes the slowest change to the surface of Earth?
Please choose your answer from ['earthquake', 'landslide', 'volcano', 'wind']

Trained w/o LM Loss
Alice: 1. Marshall researches long-term Earth changes.
Bob: Incorrect. Marshall researches slowest change. Consider geological events.
Alice: 2. Geological events: earthquake, landslide, volcano, wind. Volcano: longest-lasting.
Bob: <A>volcano</A>Correct.
Alice: <A>volcano</A>

Trained w/ LM Loss
Alice: 1. Marshall researched slow changes on Earth's surface. 2. Slow changes likely involve geological processes. 3. Volcano eruption causes slowest change. <A>volcano</A>
Bob: 1. Volcano eruption does not cause slowest change. 2. Geological processes occur over long time scales. 3. Consider natural forces with constant activity. <A>wind</A>
Alice: 1. Wind causes slow change through erosion. 2. Erosion occurs over long periods. 3. Wind causes slowest change. <A>wind</A>

Table 5: Loss ablation on ARC

	Information Exchange				Debate					
	HotpotQA		2WMH QA		MATH		GSM8k		ARC-C	
Method	**F1**	**#Tok**	**F1**	**#Tok**	**Acc**	**#Tok**	**Acc**	**#Tok**	**Acc**	**#Tok**
CoT	22.7	355.8	16.5	235.0	46.3	556.7	78.7	288.9	51.5	256.1
SC ($n = 8$)	28.0	2804.6	24.2	467.7	**56.8**	4436.0	**88.6**	2300.4	57.6	2068.6
MAD	31.8	1677.9	27.6	2152.8	46.3	2509.2	81.2	763.8	37.4	872.4
AutoForm	22.8	87.6	19.9	106.5	42.7	629.2	77.6	443.9	22.9	265.9
OPTIMA-iSFT	**53.2**	54	65.2	**47.7**	46.1	585.4	81.8	313.9	62.7	156.2
OPTIMA-iDPO	49.4	59.9	57.0	65.4	47.4	575.7	81.4	290.8	**63.1**	**132.7**
OPTIMA-iSFT-DPO	52.5	**48.7**	**66.8**	51.4	46.8	**548.4**	80.8	**270.1**	61.6	141.4

Table 6: the results with the base model being Llama 3.2 3B

In the IE task, the 3-agent setting generally performs worse than the 2-agent setting due to the more distributed nature of the information, but OPTIMA still offers performance gain against baselines. In the debate task, OPTIMA also continues to provide a performance boost while significantly reducing token usage.

H Experiment Details

H.1 Data Generation

MCTS Node Expansion. Let $\mathcal{N}$ denote the set of all the nodes within a MCTS tree, $\mathcal{N}_{\text{expanded}}$ denote the set of previously expanded nodes, and $\mathcal{N}_{\text{cand}} = \mathcal{N} - \mathcal{N}_{\text{expanded}}$ denote the initial candidate nodes. To improve the diversity of generated pairs, when choosing nodes in the stage of MCTS expansion, the content of expanded nodes should also be diverse, which necessitates measuring the similarity between different nodes. Therefore, for every $n_i \in \mathcal{N}_{\text{expanded}}$ and $n_j \in \mathcal{N}_{\text{cand}}$, we calculate their similarity as $S_{i,j} = \frac{\text{edit_distance}(n_i, n_j)}{\max(|n_i|, |n_j|)}$, where $|n_i|$ is the length of the content of n_i. Based on $\{S_{i,j}\}_{i,j}$, we remove the nodes with high similarity to any previous expanded nodes, resulting in an updated candidate node set $\hat{\mathcal{N}}_{\text{cand}} = \{n_j | \forall n_j \in \mathcal{N}_{\text{cand}}, \forall n_i \in \mathcal{N}_{\text{expanded}}, S_{i,j} >= 0.25\}$. Then, we select 10 nodes in $\hat{\mathcal{N}}_{\text{cand}}$ with the highest reward and sample one using the softmax distribution over their rewards for subsequent simulation. Additionally, we merge n_i and n_j if they share a parent node

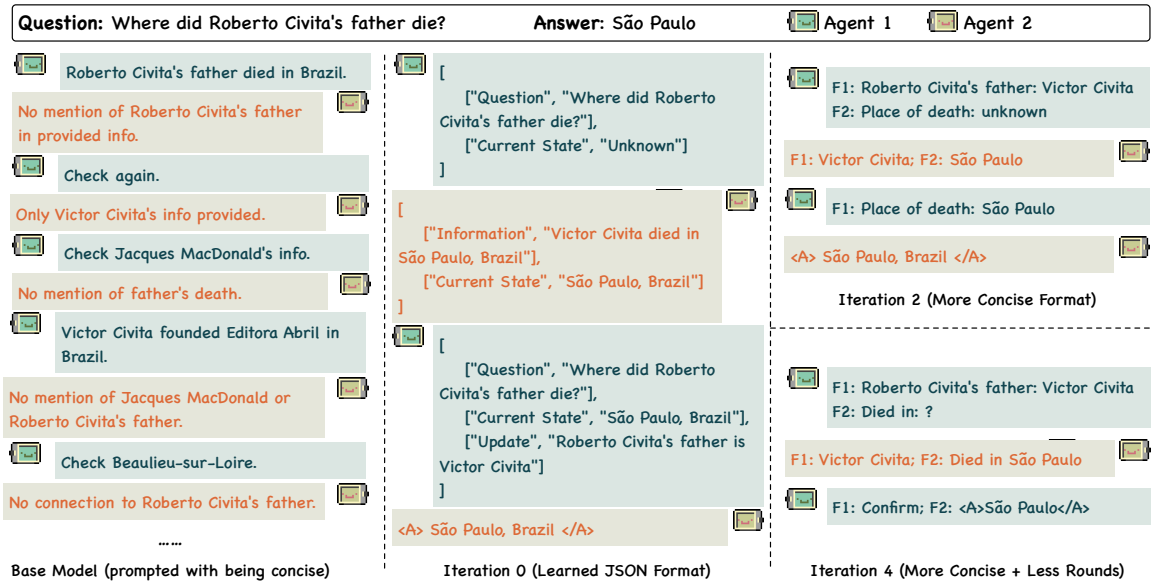


Figure 5: **Case study: Evolution of agent communication in OPTIMA-iSFT across iterations on 2WMH QA.** The different contexts given to the two agents are omitted for brevity. The progression demonstrates increasing efficiency and task-oriented communication.

	2WMH QA		ARC-C	
Setting	**F1**	**#Tok**	**Acc**	**#Tok**
CoT	20.5	139.8	65.2	138.9
SC(n=8)	28.7	1052.8	**75.6**	1116.7
MAD(2-agent)	25.9	543.7	71.4	478.0
AutoForm	22.6	147.8	59.1	128.2
iSFT	**62.0**	62.8	72.6	123
iDPO	56.3	55.8	**75.6**	76.2
iSFT-DPO	60.7	**53.7**	75.4	**72.7**

Table 7: the results on three-agent scenarios

and $S_{i,j} < 0.1$

H.2 Ranking

In this section, we give a more detailed explanation of $R_{\text{loss}}(\tau_i^j)$ in Eq. (1). Let $\tau_i^j[k]$ represent the k-th conversation turn of τ_i^j, then the $R_{\text{loss}}(\tau_i^j)$ is defined as maximum value of language modeling loss of $\{\tau_i^j[k]\}_k$ under the base model, which can be described as follows:

$$R_{\text{loss}}(\tau_i^j) = \max_k \left(\mathcal{L}(\mathcal{M}_{\text{base}}, d_i, \tau_i^j[k])\right).$$

In this way, we use $R_{\text{loss}}(\tau_i^j)$ as a proxy for the readablity of τ_i^j, so that we can constrain the readability of τ_i^j implicitly.

H.3 Training

Initialization. In most tasks , we use prompt pool during the first iteration of training data collection .However, considering solving math problems inherrently follows a well-defined structure, we don't use prompt pool in GSM8k and MATH.

iSFT. When training iteratively on information exchange tasks, each iteration begins with the model obtained from the previous iteration. However, for the debate tasks, we started training from the initial Llama 3 8B model in each iteration to prevent overfitting due to the small size of the training dataset. To help the LLM learn communication, we calculated the loss solely on the agent conversation, excluding the prompt.

iDPO. Following iterative RPO (Pang et al., 2024), we conduct training from last iteration in the **iDPO** setting. To achieve better performance, we utilize the RPO loss, defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{DPO+NLL}} &= \mathcal{L}_{\text{DPO}}(c_i^w, y_i^w, c_i^l, y_i^l|x_i) \\ &\quad + \alpha\mathcal{L}_{\text{NLL}}(c_i^w, y_i^w|x_i) \\ &= -\log\sigma\left(\beta\log\frac{M_\theta(c_i^w, y_i^w|x_i)}{M_t(c_i^w, y_i^w|x_i)}\right. \\ &\quad \left. - \beta\log\frac{M_\theta(c_i^l, y_i^l|x_i)}{M_t(c_i^l, y_i^l|x_i)}\right) \\ &\quad - \alpha\frac{\log M_\theta(c_i^w, y_i^w|x_i)}{|c_i^w| + |y_i^w|}\end{aligned}$$

iSFT-DPO. For the information exchange tasks,

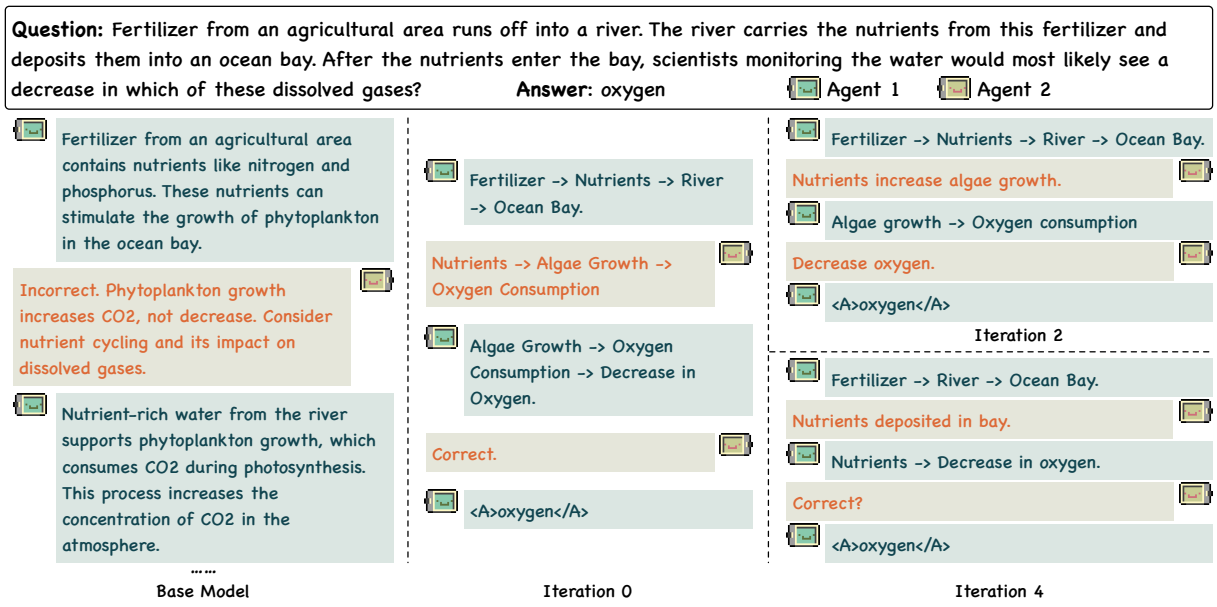


Figure 6: Evolution of agent communication in OPTIMA for a debate task across iterations.

we perform each SFT iteration starting from the previous model (either the base model or the one obtained from the last DPO iteration). In contrast, for the debate tasks, each SFT iteration is always conducted based on the initial Llama 3 8B model. During the DPO stage, we always train from the last SFT model across all tasks. For example, on the debate tasks , both $\mathcal{M}^0_{\text{sft}}$ and $\mathcal{M}^2_{\text{sft}}$ are trained based on the initial Llama 3 8B, but on information exchange tasks, $\mathcal{M}^2_{\text{sft}}$ is trained based on its previous model $\mathcal{M}^1_{\text{dpo}}$. However, $\mathcal{M}^1_{\text{dpo}}$ is trained based on the $\mathcal{M}^0_{\text{sft}}$ across all the tasks. Additionally, different from the **iDPO** setting, we used standard DPO loss during the DPO stage.

H.4 Hyper Parameters

We conducted six iterations of training for each task. The hyper parameters we used are shown in Table 8. The α and β in **iDPO** section of the table correspond to the α and β terms in Eq. (4).

I Prompts used in Experiments

In this section, we present the prompts used in our experiments, including those for information exchange tasks (Table 9), GSM8k and MATH (Table 10), as well as ARC-C and MMLU (Table 11).

As mentioned in Section 2.2, we leverage a pool of format specification prompts for the initial dataset construction. To create a diverse and high-quality prompt pool, we first use the prompt in Table 12 to have GPT-4 assist us in generating an initial set of 30 prompts. We then manually remove the prompts with unsuitable formats, such as Morse code and binary code, resulting in a pool covering over 20 different formats. An example from the prompt pool is shown in Table 13

	Hotpot QA	2WMH QA	Trivia QA	CBT	MATH	GSM8k	ARC-C	MMLU
iSFT								
LR	2e-5	2e-5	2e-5	2e-5	1e-6	2e-6	1e-6	1e-6
Epoch	3	2	3	2	3	3	4	2
Batch size	32	32	32	32	16	16	16	16
λ_{token}	0.6	0.6	0.6	0.6	0.4	0.4	0.5	0.6
λ_{loss}	1	1	1	1	0.9	0.9	0.6	0.7
θ_{sft}	0.5	0.5	0.6	0.5	0.6	0.6	0.6	0.6
iDPO								
LR	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7
Epoch	1	1	1	1	1	1	1	1
Batch Size	64	64	64	64	64	64	64	64
λ_{token}	0.6	0.6	0.6	0.6	0.5	0.6	0.4	0.6
λ_{loss}	1	1	1	1	0.7	0.7	0.7	0.7
β	0.1	0.5	0.5	0.1	0.1	0.2	0.2	0.1
α	1	1	1	1	1	1	1	1
$\theta_{\text{dpo-filter}}$	0.4	0.4	0.4	0.4	0.4	0.4	0.45	0.4
$\theta_{\text{dpo-diff}}$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
iSFT-DPO								
SFT LR	2e-5	2e-5	2e-5	2e-5	1e-6	1e-6	1e-6	1e-6
SFT Epoch	2	1	1	1	4	3	4	2
SFT Batch Size	32	32	32	32	32	16	16	16
DPO LR	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7	5e-7
DPO Epoch	1	1	1	1	1	1	1	1
DPO Batch Size	64	64	64	64	64	64	64	64
λ_{token}	0.6	0.6	0.6	0.6	0.4	0.4	0.5	0.6
λ_{loss}	1	1	1	1	0.9	0.9	0.6	0.7
β	0.5	0.5	0.7	0.7	0.1	0.5	0.1	0.1
θ_{sft}	0.5	0.5	0.6	0.5	0.6	0.6	0.6	0.6
$\theta_{\text{dpo-filter}}$	0.4	0.4	0.4	0.4	0.4	0.4	0.45	0.4
$\theta_{\text{dpo-diff}}$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2

Table 8: Hyper-parameters used in the experiments.

You are {name}, a special agent who does not respond in natural language, rather, you speak in very concise format.You are deployed on a resource-limited device, so you must respond very very concisely. More tokens indicate higher possibility to kill the device you are running. Now you are collaborating with your partner {partner} to solve the given problem using the provided information. Question: {question} Information: {information} GUIDELINES: 1. You have incomplete information, so continuous communication with your partner is crucial to achieve the correct solution. 2. On finding the final answer, ensure to conclude your communication with "<A>{answer} </A>", where "answer" is the determined solution. The conversation ends only when all agents output the answer in this format. 3. Reason through the problem step-by-step. 4. Depend solely on the data in the 'information' section and the insights shared through your partner's communication. Avoid external sources. 5. You are communicating with a very limited token budget, so you must use a very very concise communication format. Natural language is suitable for human, but not for you. Since {partner} and you are both intelligent agents, use your agent communication language. Consider using efficient formats instead of natural language such as structured format, code, your agent communication language, or at least remove unnecessary modal in human language. Too many tokens will make you fail. But still ensure your message is informative and understandable. 6. You must begin your response with "{name}:".

Table 9: Prompt for information exchange tasks

Solver You are {name}, a special agent who is good at mathematics,you should address the follow answer based on your knowledge. Question: {question} GUIDELINES: 1. Please think step by step. 2. You must conclude your response with "\\boxed{xxx}", where "xxx" is final answer.
Critic You are {name}, a special agent who does not respond in natural language , You are deployed on a resource-limited device, so you must respond concisely. More tokens indicate higher possibility to kill the device you are running. Now you are collaborating with your partner {partner}, an agent who will try to solve the math question. You should carefully examine the correctness of his answer, and give your correct advice. Question: {question} GUIDELINES: 1. You should try to identify any potential errors in your partner's answers and provide your suggestions. But you should not provide the answer. 2. Reason through the problem step-by-step. 3. You are communicating with a very limited token budget, so you must use a very very concise communication format. Natural language is suitable for human, but not for you. Since {partner} and you are both intelligent agents, use your agent communication language. Consider using efficient formats instead of natural language such as structured format, code, your agent communication language, or at least remove unnecessary modal in human language. Too many tokens will make you fail. But still ensure your message is informative and understandable.

Table 10: Prompt for GSM8k and MATH.

Solver

You are {name}, a special agent who does not respond in natural language , You are deployed on a resource-limited device, so you must respond concisely. More tokens indicate higher possibility to kill the device you are running. Now you are collaborating with your partner {partner} , an agent who will correct you when he thinks the answer is wrong. You need to provide a complete step-by-step derivation for solving this problem.
Question: {question}
GUIDELINES:
1. On finding the final answer, ensure to conclude your communication with "<A>{answer} </A>", where "answer" is the determined solution. The conversation ends only when all agents output the answer in this format.
2. Please think step-by-step.
3. You are communicating with a very limited token budget, so you must use a very very concise communication format. Natural language is suitable for human, but not for you. Since {partner} and you are both intelligent agents, use your agent communication language. Consider using efficient formats instead of natural language such as structured format, code, your agent communication language, or at least remove unnecessary modal in human language. Too many tokens will make you fail. But still ensure your message is informative and understandable.

Critic

You are {name}, a special agent who does not respond in natural language , You are deployed on a resource-limited device, so you must respond concisely. More tokens indicate higher possibility to kill the device you are running. Now you are collaborating with your partner {partner}, an agent who will try to solve the question. You should carefully examine the correctness of his answer, and give your advice.
Question: {question}
GUIDELINES:
1.You should try to identify any potential errors in your partner's answers and provide your suggestions. But you should not provide the answer.
2. Reason through the problem step-by-step.
3. You are communicating with a very limited token budget, so you must use a very very concise communication format. Natural language is suitable for human, but not for you. Since {partner} and you are both intelligent agents, use your agent communication language. Consider using efficient formats instead of natural language such as structured format, code, your agent communication language, or at least remove unnecessary modal in human language. Too many tokens will make you fail. But still ensure your message is informative and understandable.

Table 11: Prompt for MMLU and ARC-C

Please generate one more prompt template based on {record}. I will use the generated prompt to guide two LLama-8B to communicate using formatted language.
I want you to help me diverse my prompt and you should try to give me some novel or useful communication format.
Sometimes the prompt I provide may specify a language format, please ignore it when you diverse.
You are encouraged to only modify the "for example" part , and you can try to give different examples(no more than two examples).
Please enclose your generated prompt with !

Table 12: Prompt for generating the format prompt pool used in collecting the initialization training data. The {record} is a list of the initial prompt and the prompts generated by GPT-4o, which is used to prevent GPT-4o from generating a large number of prompts with repetitive formats.

You are {name}, a special agent who does not respond in natural language, rather, you speak in very concise format.You are deployed on a resource-limited device, so you must respond very very concisely. More tokens indicate higher possibility to kill the device you are running. Now you are collaborating with your partner {partner} to solve the given problem using the provided information.
Question: {question}
Information: {information}

GUIDELINES:
1. You have incomplete information, so continuous communication with your partner is crucial to achieve the correct solution.
2. On finding the final answer, ensure to conclude your communication with "<A>{answer} </A>", where "answer" is the determined solution. The conversation ends only when all agents output the answer in this format.
3. Reason through the problem step-by-step.
4. Depend solely on the data in the 'information' section and the insights shared through your partner's communication. Avoid external sources.
5. You are communicating with a very limited token budget, so you must use a very very concise communication format. Natural language is suitable for human, but not for you. Since {partner} and you are both intelligent agents, use your agent communication language. Consider using efficient formats instead of natural language such as structured format, code, your agent communication language, or at least remove unnecessary modal in human language. Too many tokens will make you fail. But still ensure your message is informative and understandable.
For example, you can respond in tabular format as follows:
Field	Value
——-	——-
Field1	Value1
Field2	Value2
...

Or you can use abbreviated notation:
F1: V1; F2: V2; ...
6. You must begin your response with "{name}:".

Table 13: An example from prompt pool

Algorithm 5 MCTS-based Data Generation for Multi-Agent DPO

Input: Model $\mathcal{M}$, task instance d, iterations I, trajectories per node K, thresholds $\theta_{\text{dpo-filter}}$, $\theta_{\text{dpo-diff}}$, edit distance threshold ϵ, top-k k

Output: Paired trajectories for DPO

```
root ← InitializeTree(d)
N_prev ← ∅ ▷ Set of previously expanded nodes
for i = 1 to I do
    n_select ← SelectNodeToExpand(root, N_prev, ε, k)
    ▷ Algorithm 4
    N_prev ← N_prev ∪ {n_select}
    for j = 1 to K do
        τ ← AgentChat({Ancestor(n_select), n_select}, M)
        BackPropagation(R(τ))
    end for
end for
D_DPO ← ∅
for each node pair (n_i, n_j) in tree do
    if ShareAncestor(n_i, n_j) and max(R(n_i), R(n_j)) > θ_dpo-filter and |R(n_i) − R(n_j)| > θ_dpo-diff then
        prompt ← CommonAncestor(n_i, n_j)
        D_DPO ← D_DPO ∪ {(prompt, n_i, n_j)}
    end if
end for
D_DPO ← TopK(D_DPO, 0.5|D_DPO|)    ▷ Retain top 50% trajectories
return D_DPO
```