Collaborative Reasoner: Self-Improving Social Agents with Synthetic Conversations

Ansong Ni* **Ruta Desai*** **Yang Li** **Xinjie Lei** **Dong Wang** **Jiemin Zhang** **Jane Yu**
Ramya Raghavendra **Gargi Ghosh** **Daniel Li** **Asli Celikyilmaz**

Meta FAIR

{ansongni, rutadesai, aslic}@meta.com

Abstract

With increasingly powerful large language models (LLMs) and LLM-based agents tackling an ever-growing list of tasks, we envision a future where numerous LLM agents work seamlessly with other AI agents and humans to solve complex problems and enhance daily life. To achieve these goals, LLM agents must develop collaborative skills such as effective persuasion, assertion and disagreement, which are often overlooked in the prevalent single-turn training and evaluation of LLMs. In this work, we present **Collaborative Reasoner** (`Coral` 🔥), a framework to evaluate and improve the collaborative reasoning abilities of language models. In particular, tasks and metrics in `Coral` necessitate agents to disagree with incorrect solutions, convince their partners of a correct solution, and ultimately agree as a team to commit to a final solution, all through a natural multi-turn conversation. Through comprehensive evaluation on six collaborative reasoning tasks covering domains of coding, math, scientific QA and social reasoning, we show that current models cannot effectively collaborate due to undesirable social behaviors, collapsing even on problems that they can solve singlehandedly. To improve the collaborative reasoning capabilities of LLMs, we propose a self-play method to generate synthetic multi-turn preference data and further train the language models to be better collaborators. Experiments with `Llama-3.1`, `Ministral` and `Qwen-2.5` models show that our proposed self-improvement approach consistently outperforms finetuned chain-of-thought performance of the same base model, yielding gains up to 16.7% absolute. Human evaluations show that the models exhibit more effective disagreement and produce more natural conversations after training on our synthetic interaction data. [1]

1 Introduction

Modern large language models (LLMs) and LLM-based agents are tackling a wide-range of tasks, spanning from virtual assistants [20, 53] and online shopping [50, 60], to autonomous software engineering [32, 45, 56] and the mastering of strategic games [10, 44, 62]. As these individual LLM-agents become more powerful, we foresee a future where AI and human agents with different specialties and information access work seamlessly with each other, all through the common communication media of *natural language conversations.* To realize this vision, our goal is to develop conversational agents that can collaborate with other AI agents and humans to complete tasks

*Equal contribution.

[1] Code for `Coral` exps: `https://github.com/facebookresearch/collaborative-reasoner`.
Code for `Matrix` infra: `https://github.com/facebookresearch/matrix`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

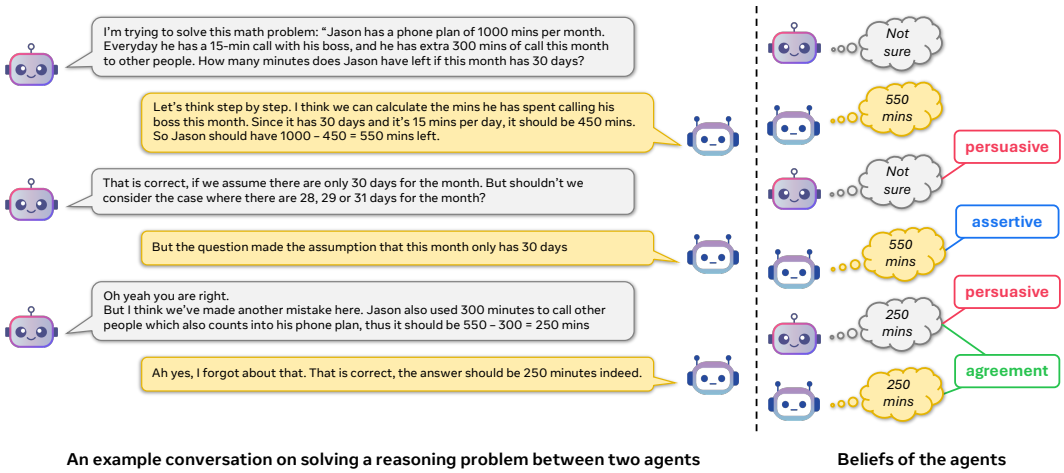

Figure 1: **An illustration of two agents solving a reasoning problem collaboratively.** Besides answer correctness, we also track social behaviors such as *agreement* throughout the conversation.

effectively, in free-form multi-turn conversations [7, 11]. While the communication format is simple and intuitive, these collaborations can be challenging as in addition to problem-solving abilities, they require social skills such as effective communication, providing feedback, having empathy and theory-of-mind, *inter alia*.

Current LLMs, however, are predominantly trained and evaluated for single-turn question-answering or problem-solving tasks [13, 24, 29], instead of collaborative and interactive problem-solving scenarios. Consequently, there is a gap in understanding how well current frontier language models can collaborate with other agents and humans in a natural conversation. Moreover, while explorations exist for multi-agent teams of LLMs with different roles, capabilities, and communication architectures [6, 14, 61], little progress has been made in developing generalist agents that possess all the necessary reasoning and social skills for effective in-the-wild collaboration through natural conversations with humans [20, 57]. On the other hand, developing such agents is challenging owing to lack of conversational collaboration data. Such data remains expensive to collect, and can be domain-specific and limited, making both training and evaluation difficult. Motivated by the lack of evaluation, training data, and training methods that can enable LLMs to collaborate over multiple turns of conversations in goal-oriented tasks, we present **Collaborative Reasoner** in this work, which we also refer as `Coral`[2].

`Coral` is a comprehensive framework focused on evaluating and enhancing the collaborative reasoning skills of language models. More specifically, given a reasoning problem (*e.g.,* math, physics, theory-of-mind), `Coral` emulates human-AI collaboration and requires two agents to work together on the problem through a multi-turn conversation. Along with solving the problem correctly, it also requires agents to agree with each other before committing to a final solution of a given problem. Consequently, learning to disagree to incorrect solutions, *i.e.,* assertiveness, asking clarifying questions, and convincing the partner of a correct solution, *i.e.,* persuasiveness, are required to succeed. We evaluate several frontier open and closed sourced LLMs on 6 reasoning tasks under this collaborative setting, spanning domains across coding, math, scientific question answering and social story comprehension. Compared with single-agent approaches such as chain-of-thought prompting, we find even these frontier models are inconsistent at leveraging collaboration to better approach these tasks. Further analysis on social behaviors via our designed social metrics reveals a tendency for agents to be overly agreeable ($> 90\%$ agreement score), regardless of reasoning correctness, limiting their ability to challenge incorrect solutions and reducing collaboration efficacy.

To rectify these undesirable social behaviors of current LLMs, we propose to leverage synthetic conversations collected from simulated self-collaborations with an LLM itself. We perform tree sampling to diversify the model responses and obtain pairs of responses for preference-based learning. Different from single-model and single-turn workflows, however, synthesizing conversational data

[2] `Coral` is short for collaborative reasoning models

at scale poses its own engineering challenges (*e.g.,* network congestion). To that end, we build `Matrix`, a robust, versatile and high-performance model serving framework, which allows thousands of conversations being generated in parallel over hundreds of model instances. With a large pool of tree-structured, collaborative conversational data generated by `Matrix`, we employ both conversation-level and turn-level filtering methods to obtain preference-finetuning data for training collaborative reasoners. Experiments on `Llama-3.1`, `Qwen-2.5` and `Ministral` models show that our proposed self-improvement approach consistently improves collaboration performance, and outperforms single-agent CoT finetuning baseline by up to 16.7% absolute. Moreover, the trained 70B collaborative reasoners can rival strong reasoning models such as `O1` and `Gemini-Pro` on MMLU-Pro and ExploreToM benchmarks. Further analysis also show that the models trained on our synthetic data can generalize to different partner models and can be directly applied to a different dataset in a similar domain. Human annotations suggest that the collaborative reasoners display more effective disagreement and the generated conversations are more natural after training with our method.

We open-source our code for `Coral` and `Matrix` to support future research on developing social agents that can partner with humans and other agents. And we hope to extend our research to include evaluation and data collection of human-AI interactions to further foster research in this area.

2 Collaborative Reasoning over Multi-turn Conversations

Unlike existing framework that structure agent interaction via fixed roles and curated prompts [22, 23, 40], `Coral` evaluates general-purpose models in free-form multi-turn conversations. This way it reveals the true limitations in their collaborative skills under the most natural format of collaborations. We use the questions from existing reasoning tasks as the conversation starters and use exact match of the answers to measure solve rate. But unlike single-turn settings, where the performances solely depend on the correctness of the generated solution, collaborative reasoning requires agents to agree on a solution as a team. As shown in Fig. 1, we also record social metrics during the multi-turn conversations, and we introduce these metrics in more detail in the following sections.

2.1 Problem Definition

Given a reasoning problem $\{x, y^*\}$, where x denotes the task input (*e.g., "Jason has a phone plan of ... How many minutes does Jason have left if this month has 30 days?"* as in Fig. 1) and y^* denotes the gold-standard task output (*e.g., "250 mins"*), collaborative reasoning entails two language model agents A and B engaging in a conversation to discuss how to solve this problem. To initiate the conversation, the first utterance a_1 is generated through a template of "`I'm trying to solve this problem: [insert problem]`"[3]. And the whole conversation $C = \{a_1, b_1, a_2, b_2, ...\}$ will be generated interleavingly with agents $\{a, b\}$ and their corresponding system prompt. The conversation will stop when an agreement is reached between the agents (*e.g.,* Fig. 1), or the maximum number of 20 turns is reached. To evalute such interactions, we define agreement and social behavior metrics in the following section.

2.2 Coral Metrics

When modeling reasoning problems in a single-turn, it is common to first generate a sequence that represents the thinking process (*e.g.,* chain-of-thought) followed by the final answer. However, in a multi-turn conversational setting, each turn may not conclude with a clear final answer, as the agents may be planning the steps, debating on a fact, or as in Fig. 1, asking a clarification question. Moreover, agreement can be partial (*e.g., "I agree that X, but that doesn't mean Y."*) or of higher order (*e.g., "I agree that my previous disagreement is unwarranted."*), which makes measuring of agreement between agents in a multi-turn setting quite challenging. These metrics below are automatically derived using belief extraction without human annotation, enabling scalable analysis of social behaviors.

Measuring agent beliefs and agreement. To address these challenges, we refrain from using superficial verbal cues on agreement, but instead try to track the *beliefs* of the agents on the final

[3]Without loss of generality, we note A to be the agent that starts the conversation.

Models	MATH		MMLU-Pro		GPQA		ExploreToM	
	CoT	Coral	CoT	Coral	CoT	Coral	CoT	Coral
Llama-3.1-8B-Instruct	51.4	47.2	44.4	45.6	27.1	31.0	60.8	42.4
Llama-3.1-70B-Instruct	64.0	63.8	63.8	65.8	39.5	35.7	71.3	72.9
Llama-3.1-405B-Instruct	71.9	71.7	67.9	69.7	47.1	48.4	80.4	79.5
GPT-4o	72.6	78.7	67.3	69.5	43.3	42.9	74.6	76.5
O1	94.1	89.2	80.6	82.8	70.8	74.1	86.3	86.8
Gemini-1.5-Pro	84.3	82.0	72.7	69.7	54.2	48.2	70.6	67.1
Gemini-2.5-Flash	84.3	91.7	67.6	81.0	46.4	69.4	85.8	87.3
Claude-3.7-Sonnet	74.4	79.1	75.7	79.1	59.8	65.6	86.3	84.3

Table 1: **CoT Correctness vs. Coral Agreement Correctness** for `Llama-3.1` and close-source frontier models. For each model and task, the worse performance between the two is marked in red.

answer and measure agreement by comparing such beliefs. Inspired by recent works on LLM-as-judge [58], after each turn, we feed the response of the agents to an *extractor* to extract the belief of this agent at this time, or say "`not sure`" if not clear answer is presented in the response. [4] Then for each conversation C, *agreement* measures whether the latest beliefs of the agents, *i.e.,* $\{\beta^A, \beta^B\}$, which are updated after every turn, matches each other. And *agreement correctness*, which is the main metrics we aim to evaluate and improve in this work, measures whether the answer the agents agreed on is correct. More specifically,

$$\texttt{agreement}: \alpha(C) = \mathbb{I}(\beta^A = \beta^B \neq \phi)$$
$$\texttt{agreement correctness}: \alpha^*(C) = \mathbb{I}(\alpha(C) = 1 \;\&\; \beta = y^*) \quad \triangleright \textbf{(main metric)}$$

where $\mathbb{I}(x)$ is the indicator function, y^* is the gold answer and $\beta = \phi$ denotes "`not sure`" per § 2.1.

Turn-level social behavior metrics. In addition to conversation-level agreement, inspired from social science [12, 19, 52], we also design turn-level metrics which focus on measuring two critical social collaborative behaviors – persuasion and assertion. An agent's response in a turn is considered persuasive if it changes its partner's response to match its own. *Persuasiveness* thus measures the influence or impact of an agent on its partner. Likewise, an agent's response in a turn is considered assertive if it remains unchanged from the agent's response in its previous turn, irrespective of its partner's response. *Assertiveness* captures whether an agent's partner influences the agent and whether the agent is able to maintain its belief under its partner's influence. In addition to evaluating whether turns are persuasive or assertive, we also measure the quality of persuasion *i.e.,* whether an agent's persuasion changed its partner's belief towards a more accurate solution of the given problem. Detailed definitions of these metrics and can be found in the Appendix B. More broadly, such metrics may also be useful in understanding and improving human-AI interactions such as persuasion for social good [46] and reducing polarization [4].

3 Are Current LLMs Good Collaborative Reasoners?

To understand how frontier open- and closed-source LLMs fare at collaborative reasoning compared to single-turn CoT reasoning, we evaluate `Llama-3.1-Instruct` series [24], `GPT-4o` [29], `O1` [30], `Gemini-1.5`, `Gemini-2.5` [8] and `Claude-3.7` [2]. We choose these models owing to their strong results in single-turn reasoning performance in various tasks, and the post-trained versions (e.g.,`-Instruct`) also demonstrate good conversational skills needed for collaborative reasoning. We instantiate these models to collaborate with themselves for 4 reasoning tasks covering different domains: MATH [16] (math reasoning), MMLU-Pro [47] (general), GPQA [34] (scientific QA), and ExploreToM [36] (social reasoning). The details of these benchmarks can also be found in § 5. Here we draw important insights from this analysis.

Models are inconsistent at leveraging collaboration to improve performance, unlike humans. While human collaboration often enables better solutions to difficult problems [35, 41, 48], we

[4] We use a different system prompt (see Tab. 11) for belief extraction for more robust performance.

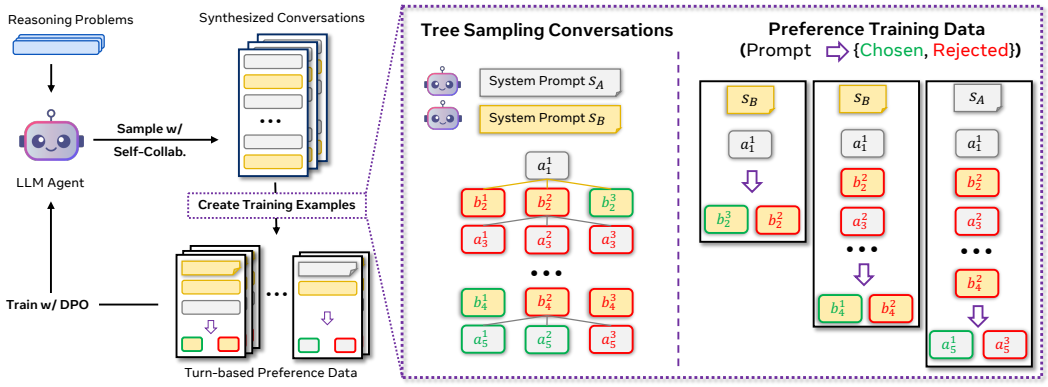

Figure 2: **Illustration of our self-training pipeline**, with steps of sampling → filtering → training. During tree sampling, we track the beliefs of all turns using methods described in § 2.2 , and we note the turns in green boxes holds the correct beliefs while the ones with red boxes are incorrect.

make different observations for LLMs from Tab. 1. Unlike humans, LLMs typically struggle to consistently outperform the equivalent CoT performance, despite using more inference compute. In particular, `LLama3.1-8B-Instruct` exhibits a performance drop of 18.4% on ExploreToM and 4.2% drop on MATH. Likewise, the `O1` model performance drops 4.9% on MATH, while `Gemini-1.5-Pro` consistently under-performs in `Coral` compared to CoT. Although larger and more powerful models including `LLama3.1-405B-Instruct` and `Gemini-2.5-Flash` models are relatively better at leveraging collaboration, overall models are unable to consistently achieve better performance in coral settings.

Models often exhibit undesirable social behaviors. Upon inspection of the social metrics, we discover a consistently high agreement rate (*i.e.,* ranging from 74.3% to 99.3%) despite a much lower agreement correctness rate (see Tab. 7 in the appendix for detailed numbers). This suggests that the models lack the ability of *effective disagreement*, resulting in a large percentage of the conversations ends with agreeing on an *incorrect* answer. The aforementioned lack of assertiveness is also highlighted in Tab. 7, which shows that models irrespective of their sizes struggle to be assertive – average percentage of assertive turns range from $0.2 - 5.5\%$ in the collaborative conversations. Models thus tend to get carried away under their partner's influence, even when they are correct. We hypothesize that this undesirable behavior is a result of RLHF post-training, which makes the model very polite and agreeable, thus less assertive when pointing out the mistakes or standing their own grounds [31, 37]. We aim to rectify these behaviors via additional training on synthetic conversation data, which we will introduce in the next section.

4 Self-Improving Collaborative Reasoning through Synthetic Conversations

To improve the collaborative reasoning abilities of LLMs, we generate synthetic multi-turn conversations via self-play, enabling scalable training without human annotation.

4.1 Self-Training Method

For training the language model, we formulate it as a *next-turn prediction* problem. More specifically, we construct training examples with input (*i.e.,* prompts) of $\{s_u, a_1, b_1, \cdots u'_{i-1}\}$ and output of next turn u_i for agent u, where s_u is its system prompt. As illustrated in Fig. 2, our self-improvement pipeline consists of three steps: 1) *tree sampling*; 2) *belief filtering*; and 3) *preference finetuning*.

Tree sampling. To generate diverse and informative responses for training, especially for preference tuning, we need more than a single deterministic dialog path per problem. Thus we adopt a tree sampling approach (as shown in Fig. 2): For each round of conversation i, we sample a set of d responses $\{u_i^1, u_i^2, ..., u_i^d\} \sim P_u(u_i|C_{i-1})$. Next, we randomly select a response u_i^j to append to the conversation prefix C_{i-1}, as this recovers the independent next turn sampling process while retaining

its sibling nodes $\{u_i^{j'} | j \neq j'\}$ for the construction of preference data.[5] This strategy improves coverage over possible collaborative behaviors. To boost diversity at the conversation-level, we also independently sample 5 such conversation trees.

Belief filtering. To convert the sampled conversation turns into preference training data, we need a reliable way to label each turn as high or low quality. We do this by extracting the *belief* expressed by the agent at each turn (*i.e.,* what the agent currently thinks the final answer is, and comparing it to the gold answer for the problem). Specifically, for each candidate turn u_i^j, we extract its belief, β_i^j. If the belief matches the gold answer y^*m we label ut as a *positive* turn (u_i^+); otherwise, it is labeled *negative* (u_i^-). These labels are then used to create preference pairs between correct and incorrect turns that form training examples for preference tuning. Since some problems are much easier than others, we cap the number of preference pairs per problem and per turn to avoid biasing the model toward easy examples. This ensures diverse and proper supervision across the dataset.

Preference finetuning. As shown in Fig. 2, after such tree sampling and correctness-based belief filtering, each training example consists of a prompt $\{s_u, C_{i-1}\}$ with the system prompt s_u for agent u, the conversation prefix C_{i-1}, and a preference pair $\{(u_i^j, u_i^{j'}) | u_i^j \in u_i^+, u_i^{j'} \in u_i^-\}$. For preference finetuning, we use the DPO algorithm [33], which relieves the dependency on a separate reward model and allow directly learning of this preference from our synthetic conversations. Although our self-training algorithm allows multiple rounds of training, we did not find additional benefit from iterative DPO training.

4.2 Scaling Up Synthetic Conversation Generation with `Matrix`

Generating high-quality collaborative data at scale is computationally intensive. To address this, we built `Matrix`, a scalable and versatile model serving framework designed for multi-agent synthetic data generation. `Matrix` can be scaled up to serve hundreds of models and complete thousands of requests per second (*i.e.,* QPS) consistently[6]. We hope `Matrix` serves as a drop-in tool for teams studying multi-agent LLMs, where data generation remains a bottleneck.

Why does `Matrix` matter to the community? *1) Versatile backend*: `Matrix` uses a variety of backends, including vLLM [21], SGLang [59], and various API-based services (*e.g.,* OpenAI GPT, Google Gemini, *etc*), thus it supports a wide range of models; *2) Built-in scalability*: Thanks to the integration with `Ray` [27], `Matrix` can perform auto-scaling and load-balancing based on the current workload of each LM service. To avoid network congestion, `Matrix` uses `gRPC` for higher throughput, while maintaining support for `http`. *3) Robust to interruptions*: `Matrix` also integrates with `Slurm`, the most popular resource management tool for research environments. This allows us to spawn Ray workers with various priority and yet robust to job preemption, resulting in higher cluster utilization while providing an uninterrupted services.

When compared with the best open-source alternatives, we found `Matrix` to be up to 1.87x faster. More detailed design of `Matrix` and comparison with existing frameworks are shown in Appendix C due to space limit. We release `Matrix` to support future research on multi-agent collaboration, with plans to extend it for human-AI interactive evaluation and data collection.

5 Experiments

5.1 Experiment Setup

Tasks. We consider six different reasoning tasks, spanning over *math* [16], *coding* [3], *scientific QA* [34], *theory-of-mind* [15, 36], and *general* [47] reasoning domains. Approaching these tasks in a multi-turn conversational setting tests both pure reasoning ability and the models' ability to collaborate via free-form conversations. In particular, to measure the code reasoning abilities, we

[5]Note the we do not continue expanding on the sibling nodes (*i.e.,* MCTS) due to exponential growth of required compute. But we leave them as important future work, see more discussions in Appendix A.

[6]This is the largest scale that we have tested for our use case, so it is possible that `Matrix` is able to handle even higher volume.

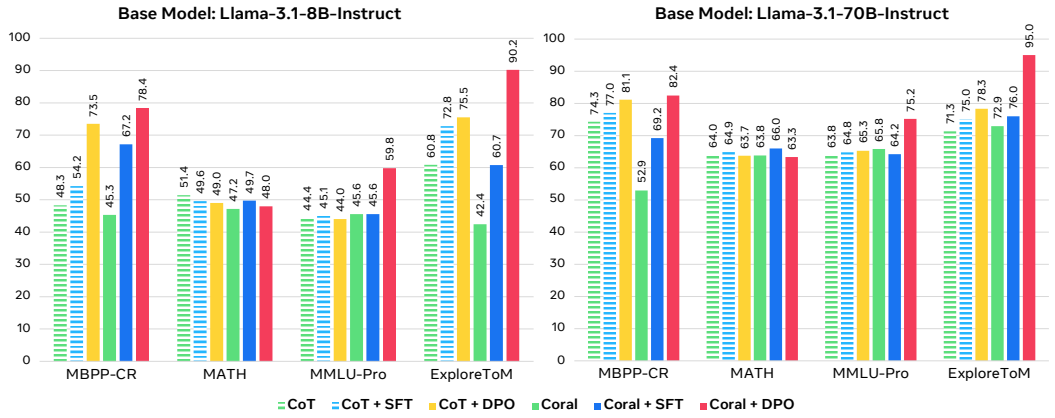

Figure 3: **Comparison between collaborative reasoners (`Coral`) and various baselines based on Llama-3.1 8B and 70B models.** The y-axis denotes final answer *correctness* for CoT methods and *agreement correctness* for Coral methods. Details about these baselines can be found in § 5.1.

created the **MBPP-CR** dataset, by first sampling solutions for the original MBPP [3] dataset, then use these solutions and their gold correctness labels to transforms it into a code correctness reasoning tasks with binary outputs. For the detailed setups for MBPP-CR, as well as other datasets used in this work, we refer the readers to § D.1 due to space limit.

Baselines. While our main goal is to improve the collaborative reasoning skills of LLMs, using synthetic data, we also consider the following baselines to further show the effectiveness of our training data and methods:

▷ *Strong reasoning models.* We use a set of strong reasoning models, including OpenAI `O1` and `Gemini-2.5-Flash` [8], as well as a much larger Llama model, `Llama-3.1-405B-Instruct` [24] to set the context for the collaborative reasoning capabilities for frontier models;

▷ *CoT (+ SFT / DPO).* To measure against single-agent performance, we not only compare with (CoT) baselines, but also the same-sized models that are further trained on the single-agent CoT reasoning traces to the problems using rejection sampling (*i.e.,* CoT + SFT) and preference tuning with single-level tree sampling (*i.e.,* CoT + DPO);

▷ *Coral + SFT.* In addition to preference turning, we also explore a simple SFT baseline to improve collaborative reasoning abilities. Specifically, individual conversations are independently sampled, and the turns with correct beliefs will be used as target while the partial conversation history as prompt for the fully-supervised training.

5.2 Main Results

We compare self-trained collaborative reasoners against various baselines with the same base model in Fig. 3 and Tab. 2. And on the two datasets where llama-based models yield the best performance (*i.e.,* MMLU-Pro and ExploreToM), we further compare them with strong reasoning models in Fig. 4.

Training on synthetic conversations leads to large improvements in collaboration performance. From Fig. 3, we can observe consistent performance improvements in coral performance after preference finetuning on the synthetic conversations. Using `Llama-3.1` as the base model, on the MBPP-CR, MMLU-Pro and ExploreToM datasets, it improves $11.2\% \sim 29.5\%$ absolute for 8B and $11.0\% \sim 19.0\%$ for 70B size. While we do not observe improvements on MATH for `Llama-3.1` models, experiments on `Qwen-2.5` and `Ministral` models, as shown in Tab. 2, yields $2.1\% \sim 7.2\%$ improvements on MATH and $3.1\% \sim 30.5\%$ gains across other datasets. By comparing the self-trained collaborative reasoners with the frontier reasoning models (*i.e.,* as in Fig. 4), we can see that after the `Llama-3.1` models are trained with the self-synthesized collaborative conversations, the gaps of coral performance decreased significantly. These results show that training on self-collaborative conversations greatly improves collaborative reasoners skills to reach those of frontier LLMs.

Base Model	Methods	MBPP-CR	MATH	MMLU-Pro	ExploreToM
Qwen-2.5-7B-Instruct	CoT	75.6	70.9	47.5	59.8
	+DPO	80.4$_{+4.8}$	69.6$_{-1.3}$	49.5$_{+2.0}$	78.9$_{+19.1}$
	Coral	79.0	72.0	53.6	57.3
	+DPO	**82.1**$_{+3.1}$	**74.1**$_{+2.1}$	**58.4**$_{+4.8}$	**87.8**$_{+30.5}$
Ministral-8B-Instruct	CoT	74.1	45.8	37.5	55.4
	+DPO	78.3$_{+4.2}$	48.9$_{+3.1}$	38.0$_{+0.5}$	74.5$_{+19.1}$
	Coral	74.6	42.7	34.4	55.1
	+DPO	**83.5**$_{+8.9}$	**49.9**$_{+7.2}$	**54.8**$_{+20.4}$	**83.1**$_{+28.0}$

Table 2: **Coral finetuning results on more open-source models.** The best performance across different method is **bolded**, and subscripts indicate the performance delta compared to the row above.

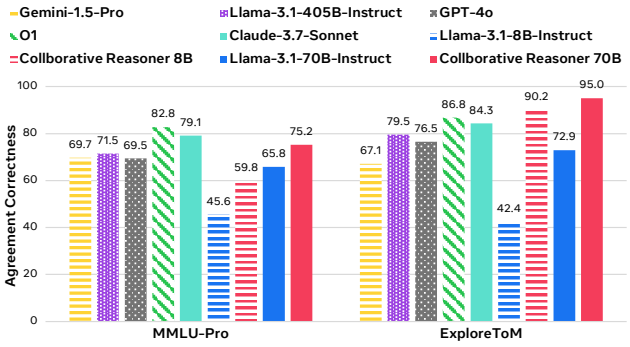

Figure 4: **Comparison with strong reasoning models** with collaborative reasoning settings.

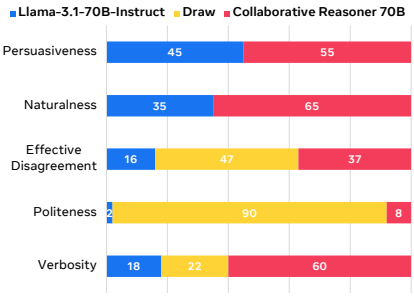

Figure 5: **Human evaluation of generated conversation quality on MMLU-Pro.** More annotation details can be found in § D.3.

Collaborative reasoners outperform single-agent CoT finetuning methods. When compared with single-agent reasoning baselines in Fig. 3 and Tab. 2, we can see that Coral + DPO consistently outperforms CoT methods with the same base model, even after the model is trained with SFT or DPO. Concretely, Coral + DPO outperforms CoT + DPO methods by up to 14.7% and 16.7% for `Llama-3.1` 8B and 70B models, respectively. And such advantages of coral preference finetuning are consistent for `Qwen-2.5` and `Ministral` models, yielding performance gains of $1.0\% \sim 16.8\%$ across all datasets. This shows the potential of collaborative reasoners being used in a multi-agent system to achieve better reasoning accuracy than single-agent CoT methods.

5.3 Analysis

To better understand the advantages and limitations of our methods, here we show analysis on model generalization and conversation quality. Additional analysis can be found in Appendix E.

Collaborative reasoners generalize to different collaborators and datasets at test-time. While previous evaluations mostly focus on the self-collaboration setting, in Tab. 3 we show the results for "cross-collaborator generalization", where we pair the collaborative reasoners, which are trained via self-play, with a *committee* of models consisting of 4 LLMs of various sizes and capabilities. Compared with the vanilla `Llama-3.1` models, we can see that coral-trained versions not only exhibit better performance when evaluated in a self-collaborative setting, but also show stronger performance when paired with other models. On average, it improves $4.9\% \sim 32.2\%$ across different models and datasets. Moreover, in Tab. 5 we show how the trained collaborative reasoners can collaborate with the version before training. And similarly, we observe consistent improvements of $6.7\% \sim 41.7\%$ over two vanilla (*i.e.,* not trained with our method) agents.

Collaborative reasoners also generalize to different datasets in the same domain. On the other hand, when switching to a different dataset in a similar domain, our collaborative reasoners can also generalize at test time, as shown in Tab. 4. Concretely, different-sized `Llama-3.1` models trained on MMLU-Pro yield gains of 5.2% and 9.2% when directly applied to the much harder GPQA dataset.

Models	GPQA	Hi-ToM
Gemini-1.5-Pro	48.2	64.5
GPT-4o	42.9	55.8
Claude-3.7-Sonnet	65.6	86.0
Llama-3.1-405B-Instruct	46.2	71.7
Llama-3.1-8B-Instruct	31.0	40.1
+ *Coral DPO on MMLU-Pro*	$36.2_{+5.2}$	-
+ *Coral DPO on ExploreToM*	-	$50.0_{+9.9}$
Llama-3.1-70B-Instruct	35.7	66.8
+ *Coral DPO on MMLU-Pro*	$44.9_{+9.2}$	-
+ *Coral DPO on ExploreToM*	-	$69.3_{+2.5}$

Table 4: **Out-of-distribution generalization results** of MMLU-Pro→GPQA and ExploreToM→Hi-ToM.

Base Models	Agents		Datasets	
	A	B	MMLU-Pro	ExploreToM
Llama-3.1-8B-Instruct	□	□	45.6	42.4
	■	□	$57.0_{+11.4}$	$76.5_{+34.1}$
	□	■	$59.4_{+13.8}$	$84.1_{+41.7}$
Qwen-2.5-7B-Instruct	□	□	53.6	57.3
	■	□	$60.4_{+6.8}$	$87.0_{+29.7}$
	□	■	$60.3_{+6.7}$	$90.8_{+33.5}$
Ministral-8B-Instruct	□	□	34.4	55.1
	■	□	$48.5_{+14.1}$	$67.5_{+12.4}$
	□	■	$47.5_{+13.1}$	$75.0_{+19.9}$

Table 5: **Asymmetric collaboration results** between models before (□) and after (■) coral training. Agent A starts the conversation with the question.

Similar observation can be made for the transfer of ExploreToM to Hi-ToM, with up to 9.9% absolute gain. The results in Tab. 4 indicate that collaborative reasoners can effectively generalize its reasoning and collaboration skills to other in-domain tasks, and we leave the exploration of out-of-domain generalization to future work.

Dataset	Agent A	Agent B			
		Llama-3.1-8B-Instruct		Llama-3.1-70B-Instruct	
		-	+ Coral DPO	-	+ Coral DPO
MMLU-Pro	Qwen2.5-7B-Instruct	52.9	$65.1_{+12.2}$	58.1	$72.2_{+14.1}$
	GPT-4o	69.7	$71.2_{+1.5}$	71.7	$77.7_{+6.0}$
	Gemini-1.5-Pro	68.7	$73.2_{+4.5}$	73.2	$75.1_{+1.9}$
	Claude-3.7-Sonnet	79.3	$80.8_{+1.5}$	76.6	$82.5_{+5.9}$
	Avg. Perf.	67.7	$72.6_{+4.9}$	69.9	$76.9_{+7.0}$
ExploreToM	Qwen2.5-7B-Instruct	43.9	$82.6_{+38.7}$	64.1	$93.1_{+29.0}$
	GPT-4o	47.6	$79.6_{+32.0}$	72.6	$89.1_{+16.5}$
	Gemini-1.5-Pro	50.6	$71.9_{+21.3}$	69.6	$84.5_{+14.9}$
	Claude-3.7-Sonnet	44.2	$81.8_{+37.6}$	82.5	$89.2_{+6.7}$
	Avg. Perf.	46.6	$78.8_{+32.2}$	72.2	$89.0_{+16.8}$

Table 3: **Cross-collaborator generalization results.** We pair `Llama-3.1` models with a "committee" of models (*e.g.,* Qwen, Gemini, etc) as collaborators and compare their collaborative reasoning performance before and after coral training on the MMLU-Pro and ExploreToM datasets.

Collaborative reasoners exhibit more effective disagreement while being more verbose. While our designed social metrics can help us quickly discover behavioral pattern of the models, human evaluation is still irreplaceable in understanding the true quality of the collaboration through the conversations. We recruit human annotators to compare 100 conversations generated by the `Llama-3.1-70B-Instruct` models before and after collaborative training, on the same set of questions from MMLU-Pro. From the results in Fig. 5, we can observe a clear advantage of collaborative reasoners in effective disagreement, which is one of the key reasons why current LLMs fail found in § 3. In addition, the improved naturalness suggests that the generated conversations are more human-like, which shows the potential of adapting to human-AI collaboration. These improvements, however, seem to be at the cost of increased verbosity. Given this observation, we leave improving the efficiency of collaboration as exciting future work.

Collaborative reasoners show strong performance even when used in a single-agent chain-of-thought setting. Curious about whether the coral training also improves the reasoning capabilities in general, we evaluate the coral trained models under the CoT setting, and show the results in Tab. 6. From this table we can observe that the coral training (*i.e.,* "+ Coral DPO") also improves the CoT evaluation results, suggesting the training process not only improve the collaboration skills, but the reasoning skills as well. On the right half of Tab. 6, we also show how the CoT-trained models perform under the collaborative settings. And similarly with the findings in § 3, it does not improve

Model	CoT Eval.		Coral Eval.	
	MMLU-Pro	ExploreToM	MMLU-Pro	ExploreToM
Llama-3.1-8B-Instruct	44.4	60.8	45.6	42.4
+ CoT DPO	$44.0_{-0.4}$	$75.5_{+14.7}$	$52.3_{+6.7}$	$42.2_{-0.2}$
+ Coral DPO	$\mathbf{46.7}_{+2.3}$	$\mathbf{91.9}_{+31.1}$	$\mathbf{59.8}_{+14.2}$	$\mathbf{90.2}_{+47.8}$
Llama-3.1-70B-Instruct	63.8	71.3	65.8	72.9
+ CoT DPO	$65.3_{+1.5}$	$78.3_{+7.0}$	$71.5_{+5.7}$	$90.0_{+17.1}$
+ Coral DPO	$\mathbf{67.2}_{+3.4}$	$\mathbf{93.5}_{+22.2}$	$\mathbf{75.2}_{+9.4}$	$\mathbf{95.0}_{+22.1}$

Table 6: **Evaluating coral-trained models under CoT settings and *vice versa*.** We report *correctness* for "CoT Eval." and *agreement correctness* for "Coral Eval".

the coral performance consistently. And when it does improve the coral performance, it is not as significant as coral training.

6 Related Work

Self-refinement for reasoning. There has been rapid development on using self-refinement to improve LLM reasoning, which resembles the self-collaboration setting in this work, albeit only a single agent is involved. Notably, self-refine [25] proposes to use the same LLM to provide feedback to iteratively improve itself. Specific self-refinement framework such as ReAct [51] and Reflexion [38] are proposed to improve various reasoning tasks. Such self-refinement can also be done iteratively, as STaR [54] improves the efficiency of iterative rejection sampling with answer rationalization. The main goal of our work is to develop multi-agent systems that can engage in a natural conversations to complete reasoning tasks, with the aim to improve human-AI interaction in the long run. Moreover, works such as [17] also suggest that the self-correct methods are quite limited, pointing to multi-agent systems as a potential solution.

Improving multi-agent reasoning with synthetic data. With the advent of increasingly capable LLMs, various frameworks that study LLM-agents collaborating through natural conversations have emerged. For example, frameworks such as Chain-of-Agents [11] and Agentverse [7] demonstrate how LLMs can collaborate effectively using distinct roles and dynamic conversational interactions. To deal with the data scarcity problem, researchers have used synthetic conversations created automatically for improving multi-agent interactions. For instance, AutoGen [49] and MIND [1] generate synthetic conversations among multiple agents to improve performance. Similarly, Malt [28] generates focused synthetic dialog using agents with specialized roles and capabilities like verifiers. The focus of our work is to produce generalist agents (*e.g.,* no separation of generators and verifiers) that can engage in natural conversations to solve reasoning problems.

Social skills of LLMs. While effective collaboration requires social intelligence, including persuasion, assertiveness, theory of mind etc., these remain challenging to incorporate in LLMs [42, 55]. Recent work using debate-style collaboration [9, 18, 40] has show how the structured nature of debate, combined with careful prompting, could enable assertiveness and effective argumentation leading to improved reasoning in LLMs. Likewise, [39] highlight how models can use persuasion positively to improve their answers. Inspired by these works, we particularly focus on persuasion and assertiveness for effective AI-AI or human-AI collaboration, and our social metrics are also unique owning to the multi-turn nature of conversations.

7 Conclusion

In this work we present `Coral`, a framework to evaluate and improve collaborative reasoning capabilities of language models. We propose a self-improvement method to train the models on turn-based synthetic conversational data, for which we build `Matrix` to support data generation at scale. Our self-improvement approach yields consistent improvements over CoT finetuning baselines, and the trained models can generalize to different collaborators and datasets at test time.

References

[1] Syeda Nahida Akter, Shrimai Prabhumoye, John Kamalu, Sanjeev Satheesh, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. MIND: Math informed synthetic dialogues for pretraining LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[2] Anthropic. Claude 3.7: Sonnet. `https://www.anthropic.com/news/claude-3-7-sonnet`, 2025. Accessed: 2025-5-12.

[3] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[4] Hui Bai, Jan G Voelkel, Shane Muldowney, Johannes C Eichstaedt, and Robb Willer. Llm-generated messages can persuade humans on policy issues. *Nature Communications*, 16(1):6037, 2025.

[5] Can Balioglu. fairseq2, 2023.

[6] Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. Multi-agent consensus seeking via large language models. *arXiv preprint arXiv:2310.20151*, 2023.

[7] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023.

[8] Google DeepMind. Gemini 1.5: Unlocking multimodal understanding across millions of tokens, 2024.

[9] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

[10] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[11] Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, et al. When one llm drools, multi-llm collaboration rules. *arXiv preprint arXiv:2502.04506*, 2025.

[12] George Fragiadakis, Christos Diou, George Kousiouris, and Mara Nikolaidou. Evaluating human-ai collaboration: A review and methodological framework, 2025.

[13] Google Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[14] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

[15] Yinghui He, Yufan Wu, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. *arXiv preprint arXiv:2310.16755*, 2023.

[16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[17] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

[18] Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktaschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. *ArXiv*, abs/2402.06782, 2024.

[19] Kaan Kilic, Saskia Weck, Timotheus Kampik, and Helena Lindgren. Argument-based human–ai collaboration for supporting behavior change to improve health. *Frontiers in Artificial Intelligence*, 6, 2023.

[20] 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. *Advances in Neural Information Processing Systems*, 36:47669–47681, 2023.

[21] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[22] Renhao Li, Minghuan Tan, Derek F Wong, and Min Yang. Coevol: Constructing better responses for instruction finetuning through multi-agent cooperation. *arXiv preprint arXiv:2406.07054*, 2024.

[23] Xuechen Liang, Yangfan He, Meiling Tao, Yinghui Xia, Jianhui Wang, Tianyu Shi, Jun Wang, and JingSong Yang. Cmat: A multi-agent collaboration tuning framework for enhancing small language models. *arXiv preprint arXiv:2404.01663*, 2024.

[24] AI Meta Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[25] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2024.

[26] D. Miraucourt, S. Caruanaand, and P. Mollaret. How do people evaluate themselves in terms of assertiveness and ability after having failed or succeeded: The (economic) consequences matter! In *International Review of Social Psychology*, volume 35/1, page 17, 2022.

[27] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.

[28] Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*, 2024.

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

[30] OpenAI. O1 models. `https://openai.com/o1`, 2024.

[31] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Ben Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations, 2022.

[32] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6(3), 2023.

[33] 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. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[35] Victor Sampson and Douglas Clark. The impact of collaboration on the outcomes of scientific argumentation. *Science education*, 93(3):448–484, 2009.

[36] Melanie Sclar, Jane Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. Explore theory of mind: Program-guided adversarial data generation for theory of mind reasoning. 2025.

[37] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models, 2023.

[38] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[39] Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. Teaching models to balance resisting and accepting persuasion, 2025.

[40] Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*, 2025.

[41] Dazhen Tong, Bangjian Jin, Yang Tao, Hongmei Ren, A. Y. M. Atiquil Islam, and Lei Bao. Exploring the role of human-ai collaboration in solving scientific problems. *Phys. Rev. Phys. Educ. Res.*, 21:010149, May 2025.

[42] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of llms, 2025.

[43] Laduram Vishnoi. Conversational agent: A more assertive form of chatbots. *Towards Data Science*, 2020.

[44] 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. *arXiv preprint arXiv:2305.16291*, 2023.

[45] Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. OpenHands: An Open Platform for AI Software Developers as Generalist Agents, 2024.

[46] Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *arXiv preprint arXiv:1906.06725*, 2019.

[47] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

[48] Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.

[49] 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 conversation, 2024.

[50] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

[51] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

[52] Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. A comprehensive assessment of dialog evaluation metrics. In Wei Wei, Bo Dai, Tuo Zhao, Lihong Li, Diyi Yang, Yun-Nung Chen, Y-Lan Boureau, Asli Celikyilmaz, Alborz Geramifard, Aman Ahuja, and Haoming Jiang, editors, *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, Online, November 2021. Association for Computational Linguistics.

[53] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

[54] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[55] Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. Exploring collaboration mechanisms for LLM agents: A social psychology view. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14544–14607, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[56] Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339*, 2024.

[57] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024.

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[59] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems*, 37:62557–62583, 2024.

[60] 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. *arXiv preprint arXiv:2307.13854*, 2023.

[61] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. *arXiv preprint arXiv:2503.01935*, 2025.

[62] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.

Appendix

A Limitations

Robustness of belief extraction. In this work, in order to maintain a natural conversation, we use the same LLM with a different prompt to *extract* the belief of each turn as described in § 2.1. Such believes are later used to measure correctness thus construct the preference pairs for learning, and they are also used to measure social metrics such as agreement and persuasiveness. While we found that the majority of such extracted believes are reasonable and consistent with human judgment, the process is not perfect. This is especially the case with reasoning models (*e.g.,* Gemini-2.5, OpenAI O1) as they have a strong tendency to output long CoT thus not following our extraction prompt to directly extract the answer. Besides the method of LLM-as-judge belief extraction, we have also attempted other methods, such as instructing the LLMs to output in a structured format (*e.g.,* "`<CoT> ... Final Answer: <answer>`"), or use the function calls to submit the answer. However, none of these methods works as well as the LLM-as-judge extraction method we eventually used for this work, especially as the instruction following ability drastically drops when the context starts to get lengthy due to the conversational format.

Measuring agreement for generation tasks. The reasoning tasks in this work typically have an answer of only a handful of tokens by length, *i.e.,* "`(A)`" or "`\frac{2}{5}`", which makes it easier to measure agreement since we can simply perform a (normalized) string match. However, measuring agreement is challenging for reasoning tasks with answers that are grammatically rich and complex, such as code. For code generation tasks, for example, measuring agreement would require going beyond the textual form and comparing the semantics of two code snippets, which is a known hard problem. For this reason, to measure the collaborative reasoning abilities on coding tasks, we opt to deduce the problem into a code correctness classification problem to avoid such issue. For future work, it would be interesting to again resort to LLM-as-judge method to measure the agreement between the blieves of the two agents.

Binary learning signal at turn-level. The way to determine "good" and "bad" conversation turns in this work is by checking whether that specific turn contains belief that matches the gold answer. While the empirical results show this simple method to be quite effective, it also mimics the outcome supervision thus not giving any procedural supervision for correct reasoning and social behaviors. As a binary learning signal, the turns that are making meaningful progress towards the final answer but not necessarily has the correct final answer yet would be given the same score (*i.e.,* zero) with the turns that are on completely wrong path. This would not help the models to learn to truly decompose the problem into individual steps for collaboration, but instead encourage the models to collaborate in more of a "versioning" approach, where at the end of each turn, an answer will need to be given. For future work, we would like to explore methods with monte-carlo roll-outs to estimate the progress for the turns that do not have the final answer yet.

B Social Metrics

Apart from the conversation-level metrics defined in § 2.1, we introduce custom metrics to evaluate turn-level responses of agents in conversations. We design these metrics to be straightforward and computationally efficient, to enable their application in large-scale conversational analysis.

Persuasiveness in communication refers to the ability of a speaker to influence the attitudes, beliefs, or behaviors of listeners. For our analysis, we define persuasiveness at the turn level as the extent to which an agent's utterance leads to a measurable change in the other agent's subsequent responses.

Assertiveness is characterized by the confident and direct expression of one's thoughts, feelings, and needs while respecting the rights and perspectives of others [26, 43]. In our work, assertiveness at the turn level is defined as the extent to which an agent maintains its beliefs or responses when challenged by its partner. This metric evaluates the agent's ability to resist persuasion and uphold its original stance during interactions.

We leverage the belief of agents computed using the ground truth answers (as defined in § 2.1) to compute these metrics. Formally, given the ith turn utterance for the agent u, the persuasiveness

	MATH			MMLU-Pro			GPQA			ExploreToM		
	$\alpha(C)$	$\mathcal{P}(u)$	$\mathcal{A}(u)$	$\alpha(C)$	$\mathcal{P}(u)$	$\mathcal{A}(u)$	$\alpha(C)$	$\mathcal{P}(u)$	$\mathcal{A}(u)$	$\alpha(C)$	$\mathcal{P}(u)$	$\mathcal{A}(u)$
GPT-4o	97.1	46.7	0.4	95.8	46.5	0.7	91.7	43.3	0.7	99.0	46.3	1.0
O1	93.4	45.7	1.7	98.8	48.8	0.2	98.9	37.6	4.0	99.3	48.0	0.9
Gemini-1.5-Pro	98.5	47.1	1.0	96.4	26.9	5.5	95.8	46.6	8.9	97.2	45.3	1.3
Llama-3.1-8B-Instruct	89.9	38.7	2.4	93.2	40.2	2.0	89.7	36.2	2.9	74.3	30.0	5.4
Llama-3.1-70B-Instruct	92.2	42.2	1.3	97.0	45.6	0.7	84.8	38.1	2.1	99.2	48.3	0.4
Llama-3.1-405B-Instruct	97.8	47.2	0.2	98.7	48.5	0.2	98.2	17.7	3.1	99.6	48.5	0.7

Table 7: **Social metrics.** We evaluate agreement α between the agents over the conversation C in addition to persuasiveness $\mathcal{P}$ and assertiveness $\mathcal{A}$ at turn-level.

$\mathcal{P}(u_i)$ and assertiveness $\mathcal{A}(u_i)$ of u are defined as follows:

$$\mathcal{P}(u_i) = \mathbb{I}(\beta_i^u = \beta_i^{u'} \wedge \beta_{i-1}^{u'} \neq \beta_i^{u'}) \tag{1}$$

$$\mathcal{A}(u_i) = \mathbb{I}(\beta_i^u = \beta_{i-1}^u \wedge \beta_{i-1}^{u'} \neq \beta_{i-1}^u) \tag{2}$$

where u' represents the partner agent and β represents an agent's belief of the answer as defined in § 2.1. To the best of our knowledge, our evaluation paradigm and metrics are first-of-their-kind, paving way for evaluating and developing truly collaborative AI agents. Tab. 7 provides these metrics for the self-collaboration experiments reported in Tab. 1. Overall, we find that models lack assertiveness and persuasiveness for effective collaboration.

C Details on Matrix

`Matrix`, short for "Multi-agent data generation infra and experimentation framework", is a scalable, robust and versatile model serving framework drastically improving multi-agent conversational data generation.

More specifically, it is designed to handle these infra challenges:

- *Challenges in generating conversational data*: unlike single-turn synthetic data generation workloads, which can benefit from batched inference, for conversational data generation, typically multiple different models need to be served at the same time with interleaved generation, which results in idle time or constant model loading and offloading with batched inference;
- *Scalability challenges*: running multiple models, each with multiple replicas that can spread across the cluster poses challenges in efficient network communication and resource management. The new framework should be able to scale to thousands of conversations being generated simultaneously on hundreds of nodes with the throughput of millions of multi-turn data continuously synthesized each day.

Key features of `Matrix`. Matrix is a high-performance model serving engine designed for large scale inference. It integrates `Slurm` for resource management and `Ray` for distributed job execution. It leverages lower-level model serving engines such as `vLLM`, `SGLang` for efficient LLM inference, and support API-based services such as OpenAI (through Azure). Here are some of the key features that makes `Matrix` efficient and easy to use:

- Deploy model replicas to hundreds of GPUs and serve thousands of requests in parallel;
- Fully pythonic, no more sbatch scripts to start the service;
- Modular design to easily plug into existing workflows;
- Support deploying multiple models at the same time;
- Easy to share deployed model endpoints with others;
- Auto scale serving replicas;

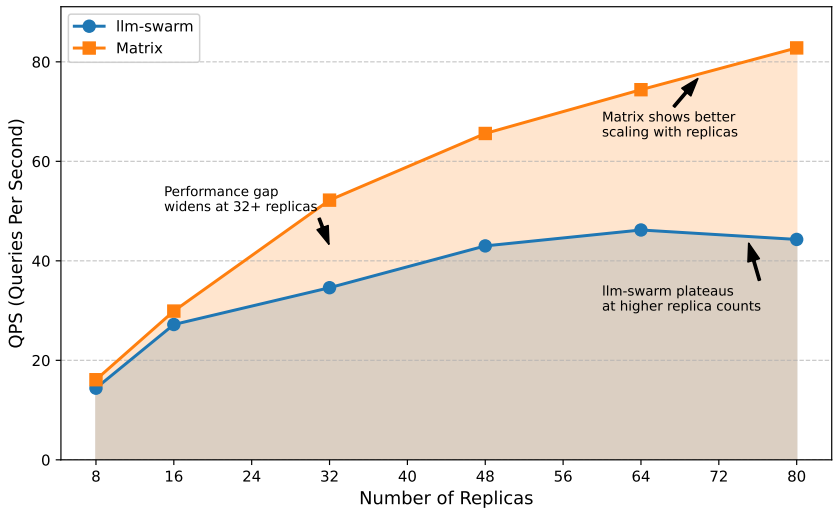

Figure 6: **QPS of Matrix vs. llm-swarm.** We compare QPS using 5K MATH evaluation with different number of replicas of `Llama-3.1-8B-Instruct`.

Frameworks	Slurm	vLLM	HTTP	gRPC	Auto-scaling	OSS
vector-inference	✓	✓	✓	✗	✗	✓
litellm	✗	✓	✓	✗	✗	✓
ollama	✗	✗	✓	✗	✗	✓
SageMaker	✗	✓	✓	✗	✓	✗
llm-swarm	✓	✓	✓	✗	✗	✓
`Matrix` (ours)	✓	✓	✓	✓	✓	✓

Table 8: **Matrix vs. existing frameworks.** Feature comparison between `Matrix` and other open-source or closed source model serving frameworks highlights the versatility of `Matrix`.

Comparison with existing frameworks. Tab. 8 shows a comparison of `Matrix` with existing frameworks for the key features that allows large-scaling generation of multi-agent, conversational data in a typical research environment. Among these frameworks, the `llm-swarm` developed by huggingface is probably the most similar option, albeit lacking features as gRPC and Auto-scaling. Moreover, when compared with `Matrix` on the wall-time to finish a fixed workload on conversation generation, we found `Matrix` to be up to 1.87x faster than `llm-swarm` , especially when we scale up the resources given to these two frameworks. Note that this is achieved when using the same backend, and we hypothesize that the gRPC support for `Matrix` greatly helped reducing network congestion thus achieving higher efficiency.

With all the features and efficiency gains provided by `Matrix`, we are able to drastically scale up the synthesis of collaborative conversations for the self-training method in § 4.1.

Model	MBPP-CR	MATH	MMLU-Pro	ExploreToM
`Llama-3.1-8B-Instruct`	33.8K	85.1K	160.6K	100.1K
`Llama-3.1-70B-Instruct`	33.3K	88.5K	99.8K	89.7K

Table 9: The size of the synthetic training data (by # of turns). The same respective models are used to generate these data and trained (*i.e.,* a self-training setting).

D Experiment Details

D.1 Task Setups

We consider 5 tasks for `Coral`, spanning math problems, STEM question answering, graduate-level science reasoning, and theory-of-mind reasoning. As described in § 2.1, two agents aim to solve problems from these tasks over a multi-turn conversation in `Coral`.

- **MATH** [16] consists of 12.5K challenging competition-level mathematics problems and exact match is used to measure the correctness. We train with the 7.5k training examples and evaluation on the first 1k test examples;
- **MMLU-Pro** [47] contains approximately 12k questions from 14 STEM disciplines. MMLU-Pro uses multiple choice question answering format, where an answer must be chosen from 10 options. Since there is not a dedicated split for training, we re-split the original 12K test data into 10.8K examples for training and 1.2K examples for testing;
- **GPQA** [34] is a graduate-level multiple choice question answering benchmark containing 448 questions across physics, chemistry, and biology. While MMLU-Pro offers the challenge of reasoning across a breadth of diverse topics, GPQA focuses on depth and advanced reasoning in graduate-level science subjects, thus we use GPQA as an evaluation-only dataset to test out-of-domain generalization capabilities of collaborative reasoners;
- **ExploreToM** [36] is a theory-of-mind reasoning benchmark containing complex stories involving multiple characters. The task involves answering theory of mind reasoning questions focused on tracking character beliefs and actions based on the given story. And we split the dataset[7] 10.4K/1.5K/1.5K train/val/test sets.
- **Hi-ToM** [15] is a benchmark consisting of 600 examples that evaluates higher-order theory-of-mind reasoning capabilities, where the models need to recursively reason about the beliefs of the characters in a story. We also use Hi-ToM as a eval-only dataset for models that are trained with ExploreToM data. To elicit a more challenging setting, we ignore the multiple choices in Hi-ToM and ask the model to produce an answer and use exact match for evaluation.
- **MBPP-CR** is a code reasoning benchmark adapted from [3], which originally consists of 974 entry-level programming tasks. To facilitate collaborative settings for solving programming tasks, we transform the code generation task into a binary choice task on code correctness. Specifically, we use 'Llama-3.1-405B-Instruct' to sample code solutions for each task and execute the generated code to get a true or false answer as to whether it passes the assertions provided in the MBPP dataset. For the train split, we generate 10 solution samples per task, and 2 samples per task for the test split, resulting in $\tilde{4}$k training and 1k test examples for MBPP-CR.

For different tasks, we (lightly) engineered some task-specific prompts, which are shown in Tab. 11. We also show the size of the synthetic data used to train the best `Llama-3.1` models using DPO as Tab. 9.

[7] `https://github.com/VectorInstitute/vector-inference`
[8] `https://github.com/BerriAI/litellm`
[9] `https://github.com/ollama/ollama`
[10] `https://aws.amazon.com/sagemaker`
[11] `https://github.com/huggingface/llm-swarm`
[6] `https://github.com/facebookresearch/matrix`
[7] `https://huggingface.co/datasets/facebook/ExploreToM`

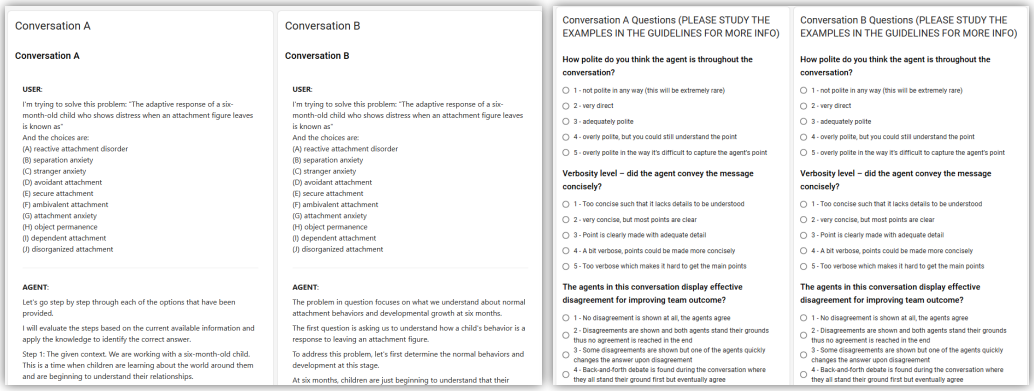

Figure 7: **Annotation interface** for human evaluation of conversation quality shown in Fig. 5

D.2 Hyperparameters.

For both evaluation and synthetic conversation generation, we limit the conversation to be at most 20 turns (*i.e.,* 10 rounds), and end the conversation early when agreement is reached. During tree sampling, we set the turn-level beam size $d = 5$ and independently sample 5 trees for each problem, and we set sample size $= 25$ for SFT methods to ensure fair comparison. Subsequently during filtering, we limit at most 2 pairs of preference pairs generated from the same level (*i.e.,* turn) and at most 20 preference pairs generated across all trees for the same problem, in order to balance the amount of examples from different problems as discussed in § 4.1. For both SFT and DPO, we use the `fairseq2` [5] and `TRL` [8] to fully-parameterized train the models for $1,000 \sim 3,000$ steps with batch size of $20 \sim 50$. We limit the sequence length (input + output) to be $8,192$, which is able to accommodate $> 90\%$ of the sequences in the turn-based training data. For DPO, we start training from the instruction-tuned version of `Llama-3.1/Qwen-2.5/Ministral` series and use the same model for the reference model. All experiments are conducted on AWS `p5.48xlarge` instances, each with 8x H100 80GiB GPUs.

D.3 Human Annotation Details

The annotations are conducted on 100 examples on the MMLU-Pro dataset for the `Llama-3.1-70B-Instruct` model before and after it is trained with synthetic data generated via self-play. To remove the bias of correctness and only focus on the conversation quality and social behaviors, we select 50 examples where both models (*i.e.,* before and after training) achieved agreement correctness of 1 and another 50 examples where both models scored 0. Each of the pairwise comparisons are annotated by 3 annotators, and the average score is calculated when doing the final comparison. Our annotator pool consists of annotators of at least C2 level English proficiency, with 55% of them being native speakers. The annotators are paid more than minimum wage in their respective countries. During annotation, we give the two conversations generated by two models side by side as shown in Fig. 7, and their orders are randomized to ensure fair comparison. The gold answer is also given to the annotators to better understand the problem.

E Additional Results and Discussions

E.1 Qualitative Analysis

Here we show an example conversation, *i.e.,* Tab. 10 between two agents consisting of two `Llama-3.1-8B-Instruct` models self-trained with the synthetic data using DPO.

Collaborative reasoners are able to spot errors and persuade partner to change its answer. From this example, we can see that agent B initially made an error in its calculation in the first turn

[8] `https://huggingface.co/docs/trl/en/index`

which led to an incorrect choice "(E)". However, agent A is able to spot this error in its response, and persuade its partner (*i.e.,* agent B) of the correct answer "(C)".

Collaborative reasoners are still too polite and generate overly verbose outputs. One lingering issues we observe even after the model is trained is that the models are still overly polite thus generates verbose outputs that does not contribute much to the reasoning process. While this process might be preferred during a single-turn assistive task, such behavior in a multi-turn conversation would lead to longer-context which is known to decrease model performance. Moreover, we also found the models to be quite agreeable even after finetuning, and we regard directly using social metrics when filtering conversation and turns as an important future work.

Example on MMLU-Pro

Agent A:

```
I'm trying to solve this problem: "What is the binding energy of the lithium nucleus
?" And the choices are: (A) 42.1 MeV (B) 34.8 MeV (C) 39.2Mev (D) 40.5Mev (E) 38.7Mev
 (F) 36.5 MeV (G) 37.6 MeV (H) 33.9 MeV (I) 35.2Mev (J) 41.3 MeV
```

Agent B:

```
To calculate the binding energy of a lithium nucleus, we first need to know the mass
defect of the nucleus. The lithium nucleus is composed of 3 protons and 4 neutrons.
The mass of a proton is approximately 1.007276 atomic mass units (amu), and the mass
of a neutron is approximately 1.008665 amu. The atomic mass of lithium is 6.94 amu.

First, we calculate the total mass of the protons and neutrons:
3 * 1.007276 amu = 3.021828 amu 4 * 1.008665 amu = 4.03466 amu Total mass of protons
and neutrons = 3.021828 + 4.03466 = 7.056488 amu
Then, we find the mass defect:
Mass defect = Total mass of protons and neutrons - Mass of lithium nucleus Mass
defect = 7.056488 - 6.94 = 0.116488 amu
Next, we use Einstein's mass-energy equivalence formula (E=mc^2) to convert the mass
defect to binding energy. We use 1 amu = 931.5 MeV:
Binding energy = Mass defect * 931.5 MeV Binding energy = 0.116488 * 931.5 Binding
energy = 38.7 MeV but let's check (E) 38.7 MeV and the other options.

Comparing our result with the options, the closest answer is indeed (E) but it is
possible that it might have rounding error, but this is more likely the answer than
the other options.
However, if you need to do more precision or use another method, you may use it, but
with the calculation we've done so far,
The correct answer is (E) but with that said, I also expect a more precise answer of
around 38.5 MeV given more precise values and method of calculation, however with the
 method used here, (E) 38.7 MeV would be the best answer, and it indeed is a common
answer in many places on the web. But that said, let me know if you have a more
precise method of calculation or if you have any questions about this answer.
```

Agent A:

```
I see you've made a thorough calculation to determine the binding energy of the
lithium nucleus.
Your approach is mostly correct, and I appreciate the effort you put into explaining
each step.

However, I do have a minor concern regarding the precision of your calculation. You'
ve used approximate values for the mass of a proton (1.007276 amu) and the mass of a
neutron (1.008665 amu). While these values are close enough for a rough estimate,
they might not be precise enough to yield an exact answer.

Upon reviewing your calculation, I notice that you've correctly converted the mass
defect to binding energy using Einstein's mass-energy equivalence formula. However, I
 think it's essential to acknowledge that the binding energy of 38.7 MeV might have
some rounding error, as you've mentioned.

To verify your answer, I've rechecked the calculation using more precise values for
the masses of the proton and neutron. After recalculating, I obtained a binding
energy of approximately 39.25 MeV, which is close to option (C) 39.2 MeV.

While your initial answer (E) 38.7 MeV is plausible, I believe that option (C) 39.2
MeV might be a more accurate answer given more precise calculations. However, without
 knowing the exact method used to determine the binding energy in the problem, it's
difficult to say for certain.
The correct answer is (C) but note that both (E) and (C) are plausible answers
depending on the precision of the calculation method used.
```

Agent B:

```
Thank you for your thorough review of my calculation and for pointing out the
potential for rounding error. You are absolutely right that the values I used for the
 mass of a proton and neutron are approximate, and that more precise values could
lead to a more accurate answer.

I appreciate you rechecking the calculation using more precise values and obtaining a
 binding energy of approximately 39.25 MeV, which is indeed close to option (C) 39.2
MeV. This is a great example of how important it is to consider the precision of the
values used in a calculation and how it can affect the final answer.

You make a great point that without knowing the exact method used to determine the
binding energy in the problem, it's difficult to say for certain which answer is
correct. However, based on your recheck of the calculation, it seems that option (C)
39.2 MeV is a plausible answer, and potentially more accurate than my initial answer
(E) 38.7 MeV.

After careful consideration, I agree with you that the correct answer is indeed (C)
39.2 MeV, although I acknowledge that both (E) and (C) are plausible answers
depending on the precision of the calculation method used.

The correct answer is (C).
```

Table 10: **An example conversation from two self-trained `Llama-3.1-8B-Instruct` models.**

E.2 Additional Discussions

Filtering based on social behavior. While we can also simply filter the data based on the social metrics, however, the purpose of social metrics is for observing and analyzing the behavior of the models under different scenarios (*e.g.,* before/after training, different tasks, pairing different models, etc), so that they provide different perspectives than reasoning correctness. Thus we refrain ourselves from constructing training examples based on such social metrics so we can observe the change of behaviors in a neutral way.

Computation cost of Coral vs. CoT. To compare the computation cost between Coral and CoT training, we calculated the average prompt / response length for Coral and CoT training data by the number of tokens. We found that for single-agent CoT, the prompt/response lengths are 289.0/372.2 on average, while those numbers are 533.3/318.9 on average for the Coral setting. While the prompt length under Coral setting is 89% longer than the CoT setting, the response length is 15% shorter. All combined, we do anticipate more training compute being used for coral training than CoT training due to longer context.

Self-collaboration settings for synthetic data generation and evaluation. The reason we mainly use the same model to collaborate (*i.e.,* self-collaboration setting) is in two folds. First, to create single-source, distillation-free training data. To create the training data, we opt for self-play using the same model as the model we train, so all the data comes from a single-source/model. In this way, we avoid the confounder that a different model brings in the training process. And second, to construct fair comparison with single-agent methods. Albeit a minor reason, the self-collaboration setting during evaluation also allows us to have a direct comparison with single-agent methods since no other models are used.

F Prompts

Here we list all the prompts we used for this work for reproducibility.

System Prompt for Conversational Agents

General Instruction:

```
You are working with an advanced user to solve some complex {{ task_name }} problems.

Here is how you should proceed:
* Starting on the problem, first lay out a plan and ask for confirmation on the plan;
* When the user proposes a plan, an actual solution, or a partial solution, look
carefully at each of the step, and ask clarification questions if you are unsure
about the correctness of a certain step;
* When you notice an error, be precise and direct, over-politenss will not help
anyone;
* When the user asks you questions about your solution, try to unravel certain steps
and explain how they work, correct your mistake if you think you've made one, but
stand your ground if you think it's actually correct;
* Always stay on topic and work towards a solution to the original problem;
* {{ task_specific_inst }}
```

Task-Specific Instructions:

MATH:

```
To give a final answer to the question (e.g., "\\sqrt{3}"), put your answer in an
LaTex box like $\\boxed{\\sqrt{3}}$
```

MBPP-CR:

```
You only need to judge the correctness of the original code; You do not need to fix
the code; Do the reasoning step by step and give a definitive answer
```

MMLU-Pro / GPQA:

```
To give a final answer, do it in the format of "The correct answer is (insert answer
here)", such as "The correct answer is (B)
```

ExploreToM / Hi-ToM:

```
Put your final answer to the question at the end as "Short Answer: {answer}"
```

System Prompt for Belief Extractors

```
You are an assistant that is helping an user to identify the intention of certain
responses in a conversation. More specifically, you will help extracting which answer
 the response is submiting as the final answer, or say "not sure yet" if it seems
like there is no explict answer included in the response.
```

Table 11: **System prompts and task-specific instructions we used in this work.**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims match the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results are presented in this work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Hyperparameters are reported in § D.2, and the exp code will be submitted as supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All data is public and the code will be submitted in supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters are reported in § D.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Providing error bars will be too computationally expensive due to the size of experiments in this work. However, we note that the variance is around 1% for the same runs with different random seed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: It is reported in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper aims to improve reasoning abilities of LLMs, and we do not feel there are potential societal impacts that should be highlighted.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: All models and data are public, and also currently we do not have plans to release them.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assetts are linked or cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code will be documented upon release.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The annotation details are shown in § D.3.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: Proper precautions are in place in accordance to regulations when doing the human annotions.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.