Large Language Model-based Human-Agent Collaboration for Complex Task Solving

Anonymous ACL submission

Abstract

In recent developments within the research community, the integration of Large Language Models (LLMs) in creating fully autonomous agents has garnered significant interest. Despite this, LLM-based agents frequently demonstrate notable shortcomings in adjusting to dynamic environments and fully grasping human needs. In this work, we introduce the problem of LLM-based human-agent collaboration for complex task-solving, exploring their synergistic potential. In addition, we propose a **Re**inforcement Learning-based **H**uman-**A**gent **C**ollaboration method, **ReHAC**. This approach includes a policy model designed to determine the most opportune stages for human intervention within the task-solving process. We construct a human-agent collaboration dataset to train this policy model in an offline reinforcement learning environment. Our validation tests confirm the model's effectiveness. The results demonstrate that the synergistic efforts of humans and LLM-based agents significantly improve performance in complex tasks, primarily through well-planned, limited human intervention. Datasets and code are available at: `https://anonymous.4open.science/r/ReHAC`.

1 Introduction

In today's increasingly complex world, humans are confronted with multifaceted tasks stemming from technical, social, and economic domains. Solving these complex tasks necessitates not only human interaction with the environment but also intricate decision-making processes. To alleviate human workload and enhance the automation of tasks in both professional and personal spheres, researchers have been actively developing advanced tools for human assistance (Zawacki-Richter et al., 2019; Amershi et al., 2019). Recently, the emergence of Large Language Models (LLMs) such as LLaMA (Touvron et al., 2023), Gemini (Team et al., 2023) and GPT (Brown et al., 2020; Achiam et al., 2023) has marked a significant milestone. LLMs' remarkable abilities in task understanding, planning, and reasoning (Zhao et al., 2023b) have given rise to the development of LLM-based autonomous agents (Wang et al., 2023a; Yao et al., 2022; Shinn et al., 2023). These agents are designed to leverage the LLMs' capabilities to assist humans in solving complex tasks autonomously. The LLMs' capabilities enable them to effectively navigate and address the complexities encountered in real-world scenarios, thereby offering substantial support in human decision-making processes of task-solving.

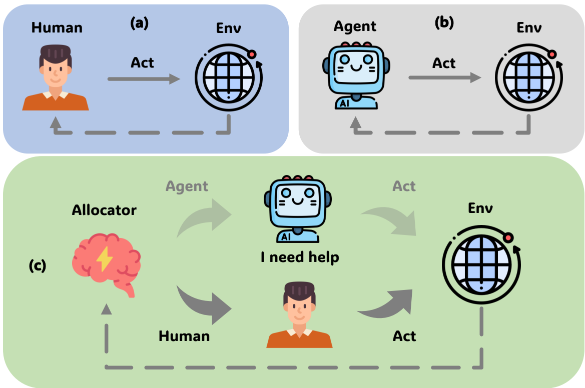


Figure 1: Different Levels of Automation. **(a) No automation:** Tasks are entirely performed by humans. **(b) Full automation:** Tasks are completely executed by agents without human intervention. **(c) Conditional automation:** Humans are required only for specific subtasks, without continuous monitoring.

Despite the remarkable progress of LLM-based agents, there remains a notable gap in their intelligence level to handle complex and dynamic real-world tasks with human-like proficiency. This limitation poses a significant challenge to their practicality in real-world applications, especially in scenarios where high accuracy is crucial, such as the legal or financial domains. Addressing this challenge extends beyond just enhancing the agents' capabilities. Incorporating human intuition and wisdom is equally vital for the effective management of these intricate and evolving tasks, offering a complementary approach to the limitations of current agent technologies.

In this work, we introduce the problem of **LLM-based human-agent collaboration for complex task solving**, aiming to augment the capabilities of LLM-based agents by integrating human intuition and wisdom. The idea is analogous to the evolution in autonomous driving technology, which has been categorized into varying levels of autonomy, ranging from no automation, conditional automation to full automation (Khan et al., 2022; SAE International, 2021). Referring to this framework, we define the different levels of human-agent collaboration, as illustrated in Figure 1. Applying this conditional automation mode to LLM-based agents offers a practical path for their deployment in real-world scenarios, acknowledging the current limitations in their cognitive capabilities. Instead of aiming for full automation, human-agent collaboration under the paradigm of conditional automation enables humans to intervene the complex task-solving when necessary, while agents handle most of the sub-tasks. This takes advantage of both human and machine intelligence.

While advancements in LLMs significantly enhance the capacity for mutual understanding in human-agent collaboration, several crucial challenges persist. These challenges include defining the division of labor between humans and agents, determining the granularity of tool execution, managing proactive interruption, and implementing multi-level intervention. However, our research specifically focuses on scenarios where humans directly replace agents in action. The key challenge we aim to address in human-agent collaboration lies in determining the optimal stages for human intervention in task-solving and minimizing such intervention to enhance efficiency. Some researchers have made preliminary attempts, by designing heuristic rules or specialized prompts to determine the stages at which agents should seek human assistance (Cai et al., 2023; Wu et al., 2022a; Mehta et al., 2023; Wang et al., 2023b). However, these rule-based or prompt-driven approaches are heavily reliant on specific application contexts and lack universality. They often demand a deep understanding of the domain and substantial experience from the designers, otherwise, suboptimal design choices can lead to reduced performance. Apart from that, a standardized formal framework and universally accepted paradigm for leveraging large language models (LLMs) in human-agent collaboration is still lacking.

To overcome the aforementioned challenges, we propose a **Re**inforcement Learning-based **H**uman-**A**gent **C**ollaboration method, **ReHAC**, aimed at effectively combining human intervention with the automation capabilities of LLM-based agents. Our method, leveraging reinforcement learning, trains a policy model to dynamically identify the most advantageous moments for human input during the task-solving process. **ReHAC** is a learnable general framework that can be applied to various scenarios and does not require additional prior knowledge to design rules and prompts. For training this policy model, we collect a dataset comprising tasks collaboratively completed by humans and LLM-based agents, utilized for the offline training of the policy model. We conducted extensive experiments on three multi-step reasoning datasets: HotpotQA, StrategyQA, and InterCode, using two popular LLM-based agent frameworks, ReAct and "Try-again". The experimental results indicate that with a policy model learned from limited data, ReHAC can effectively allocate human intervention in human-agent collaboration scenarios, thereby achieving a balance between effectiveness and efficiency.

2 Approach

In this section, we first formulate the problem of human-agent collaboration for complex task solving, and then introduce our proposed ReHAC method in detail.

2.1 Preliminary and Problem Formulation

Complex task-solving, inherently necessitating multi-step planning and reasoning, is conventionally formalized as a multi-step decision-making problem. Historically, complex task-solving was predominantly achieved through **human-driven methods**. These methods leveraged human cognitive capabilities to determine the suitable action in each step. Formally, considering a complex task q, it is traditionally solved via a sequence of actions $(a_1, a_2, \cdots a_n)$, with each action determined by human decision-making, expressed as:

$$a_t = \text{Human}(q, s_t), \tag{1}$$

where $s_t = (a_1, o_1, \cdots, a_{t-1}, o_{t-1})$ denotes the history information of task state at step t and o_t is the observation after a_{t-1} is proceeded.

The advent of LLMs has brought a paradigm shift in this arena. Their impressive understanding and reasoning abilities have prompted research into LLM-based agents for complex task-solving, thereby enhancing the level of automation in task-solving. These **agent-driven methods** (e.g., ReAct (Yao et al., 2022)), leverage LLM-based agents to supplant human decision-making. This shift is represented as:

$$a_t = \text{Agent}(q, s_t). \tag{2}$$

This evolution of such AI-driven techniques provides a way to the automation of complex task-solving.

However, limited by the current intelligence level of LLMs, full automation based on agent-driven methods is not yet feasible in practical scenarios (Kiseleva et al., 2022; Mehta et al., 2023). Inspired by autonomous driving (Cui et al., 2024; Fu et al., 2024; Bastola et al., 2024), we propose the problem of **LLM-based human-agent collaboration for complex task solving** and explore the dynamics and efficacy of the **human-agent collaborative methods** for complex task solving. We first explore a specific form of human-agent collaboration: humans intervene in the complex task-solving process when necessary. Formally, we need to determine whether a human or an agent makes decisions based on the actions' complexity and contextual changes, i.e.,

$$a_t = \text{Human}(q, s_t) \quad \text{or} \quad \text{Agent}(q, s_t), \tag{3}$$

It is generally perceived that direct human intervention in decision-making, particularly in real-world scenarios, incurs higher costs and diminishes the system's automation level (Cai et al., 2023; Wang et al., 2023b). On the other hand, human intervention plays an important role in enhancing task performance and flexibility. Therefore, the objective of human-agent collaboration is to enhance the effectiveness of complex task-solving with minimal reliance on human decision-making. One key challenge is to **determine the stages in the task-solving process where human intervention is most beneficial and effective, aligning with the goal of minimizing human involvement while maximizing task performance**.

2.2 ReHAC

In this work, we propose a Reinforcement learning-based Human-Agent Collaboration method, ReHAC. It formulates the human-agent collaboration problem as a Markov Decision Process (MDP) framework, represented by the tuple $(S, \mathcal{A}, P, R, \gamma)$, where S is the set of states, $\mathcal{A}$ is the set of actions, $P : S \times \mathcal{A} \times S$ is the state transition probabilities, R serves as the reward function, and γ the discount factor.

For each action $a_t \in \mathcal{A}$, we define it as a tuple $(a_t^{collab}, a_t^{task})$, where a_t^{collab} indicates the subtask is allocated to an agent or a human, and a_t^{task} is the task action determined by agent or human:

$$a_t^{collab} \sim \pi_{\theta_1}^{collab}(a_t^{collab}|s_t)$$

$$a_t^{task} \sim \begin{cases} \pi_{\theta_2}^{task}(a_t^{task}|s_t), & \text{if } a_t^{collab} = 0; \\ \pi_{\text{Human}}^{task}(a_t^{task}|s_t), & \text{otherwise}, \end{cases} \tag{4}$$

where $\pi_{\theta_1}^{collab}$ is the collaboration policy model, $\pi_{\theta_2}^{task}$ is the agent-based task policy model, and $\pi_{\text{Human}}^{task}$ is the human task policy.

To balance the maximization of task performance and the cost of human intervention, we define the reward function as:

$$R(s, a) = T(s, a) - \lambda C(s, a), \tag{5}$$

where $T(s, a)$ is the measure of expected task rewards received after taking action a in state s, $C(s, a)$ is the number of human interventions in the trajectory after taking action a, λ is a hyper-parameter that serves as a penalty coefficient of the number of human interventions. We utilize Monte-Carlo estimation to compute this reward function.

Optimization: Following the REINFORCE algorithm (Williams, 1992), we optimize the expected reward:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{\pi_\theta}[R(s, a)], \tag{6}$$

which aims to find an optimal policy π_θ that ensures the maximization of task rewards while minimizing the human intervention costs, and $\theta = [\theta_1, \theta_2]$. We utilize the advantage function to enhance the stability of optimization and important sampling for offline learning:

$$A(s, a) = R(s, a) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} R(s, a')$$

$$\nabla_\theta \mathcal{J}(\pi_\theta) = \sum_s \sum_a w(s, a) \nabla_\theta \log \pi_\theta(a|s) A(s, a),$$

$$w(h, a) = \text{Clip}\left(\frac{\pi_\theta(s, a)}{\pi_{\text{beh}}(s, a)}\right), \tag{7}$$

where $A(s, a)$ is the advantage function, the clip function limits the importance sampling term to the interval $1 - \epsilon$ to $1 + \epsilon$, and the behavior policy π_{beh} represents the policy under of the offline training. Moreover, we have incorporated an entropy regularization term. This term encourages the policy to explore a variety of actions, thereby preventing the policy from becoming too deterministic and overfitting to the training data. Finally, the gradient of objective function is as follows:

$$\nabla_\theta \tilde{\mathcal{J}}(\pi_\theta) = \nabla_\theta \mathcal{J}(\pi_\theta) + \alpha \nabla_\theta H(\pi_\theta(\cdot|s)). \quad (8)$$

3 Experiments

3.1 Experimental Setup

Datasets Following Yao et al. (2022); Shinn et al. (2023); Liu et al. (2023b); Xu et al. (2023), we evaluate the efficacy of our method on question answering and coding datasets: (1) HotpotQA (Yang et al., 2018) is a Wikipedia-based question answering benchmark which needs model to perform multi-hop reasoning over complex questions. (2) StrategyQA (Geva et al., 2021) is a question answering benchmark with questions that need implicit reasoning. (3) InterCode (Yang et al., 2023) is an interactive coding dataset that enables agents to receive feedback from the code interpreter. In this work, we use InterCode-SQL part, which requires models to write SQL statements to fulfil the query.

Implementation details We use LLaMA-2 (Touvron et al., 2023) as the collaboration policy model $\pi_{\theta_1}^{collab}$ and use Low-Rank Adaptation (LoRA, Hu et al. (2021)) methods to train the policy model. In all experiments, we utilized ChatGPT (gpt-3.5-turbo-0613) to simulate the agent policy $\pi_{\theta_2}^{task}$. More model implementation and data collection details can be found in Appendix A.1.

In this study, we set humans and agents to solve tasks under the ReAct framework (Yao et al., 2022) for question-answering datasets. The action space of a^{task} is {Search[entity], Lookup[keyword], and Finish[answer]}. All actions are supported by a Wikipedia web API, following the original ReAct implementation. For the InterCode dataset, we solve tasks under the “Try Again” framework (Yang et al., 2023). Here, agents and humans interact with the code interpreter through the action a_t and receive execution outputs from the code interpreter as observations o_t. The task-solving process ends if any one of the following conditions is satisfied: 1) the Finish[answer] action is executed actively by $\pi_{\theta_2}^{task}$ for the question answering dataset. 2) the task reward $T(s, a) = 1$ for InterCode dataset. 3) the number of actions t exceeds a pre-defined step threshold.

Reward Calculation For all datasets, the final reward is computed as equation (5). For question answering datasets, we choose the F1 score as the task reward $T(s, a)$. For the InterCode dataset, following Yang et al. (2023), we use Intersection over Union as the task reward $T(s, a)$.

Baselines We compare our method ReHAC with the following baselines: 1) Agent-only which carries out all actions by agents. 2) Human-only, which conducts all actions by humans. 3) Random, which selects an agent or human randomly at a probability of 50% to perform each action. 4) Prompt, which prompts the agent to actively decide whether the action is executed by itself or a human. 5) Imitation Learning (IL), which trains the policy model to decide whether the action should be finished by an agent or human by the IL method. More details about baselines can be found in the Appendix A.2.

3.2 Overall Results

In this section, we verify the effectiveness of our proposed ReHAC method for human-agent collaboration on the HotpotQA dataset.

Human-Agent Experiments Figure 2(a) shows the evaluation results of human-agent collaboration on the HotpotQA dataset. From the figure, we can observe that all human-agent collaboration methods outperform Human-only and Agent-only methods. This underscores the importance of collaborating human and agent in complex task-solving for getting higher reward. In addition, ReHAC$_{\text{Human}}$ achieves the best performance compared with prompt-based and random-based method in achieving higher rewards. Specifically, when $\lambda = 0.06$, ReHAC achieves a higher reward with approximately 30% more human interventions compared with the prompt-based baseline; when $\lambda = 0.1$, it also achieves a reward improvement with about 20% less human interventions. This indicates that our ReHAC method can dynamically introduce human intervention in real human-agent collaboration scenarios, thereby achieving a balance between effectiveness and efficiency.

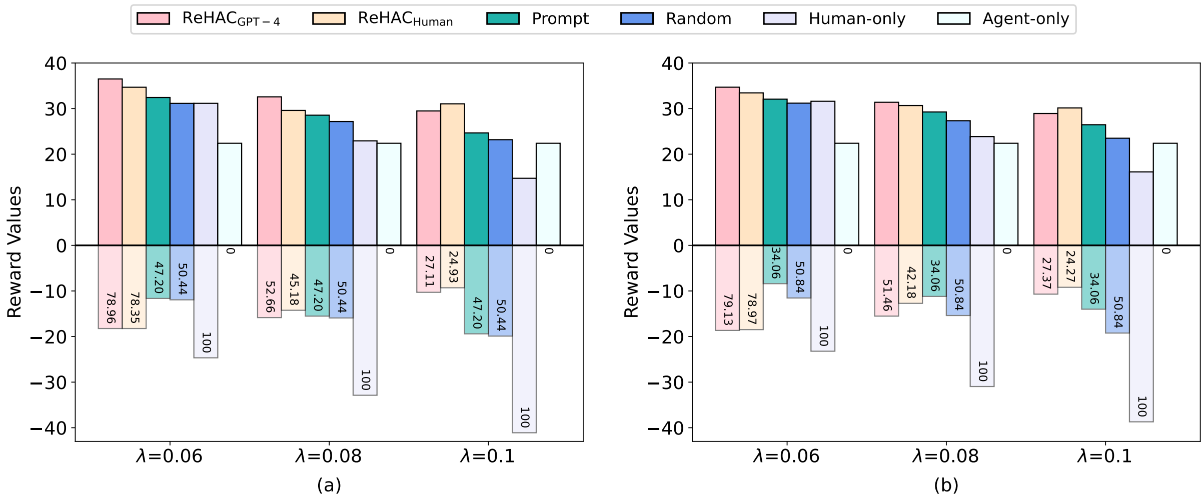


Figure 2: (a) Human-agent collaboration evaluation. (b) GPT-4-agent collaboration evaluation. The bars above the 0-axis represent the reward R, the bars below the 0-axis represent the human intervention cost λC, and the entire columns, composed of the bars above and below the 0-axis, represent the task reward T. Numbers within the bars means the human intervention rate (%). ReHAC$_{\text{GPT-4}}$ and ReHAC$_{\text{Human}}$ represent the policy model trained on GPT-4-agent and human-agent collaboration datasets, respectively. ReHAC outperforms other baselines in human-agent collaboration scenarios.

Focusing on ReHAC$_{\text{Human}}$, we observe that as λ increases, the human intervention rate[1] (HIR) of ReHAC$_{\text{Human}}$ gradually decreases. This trend suggests that a higher human penalty coefficient elevates our policy model's "threshold" for assigning actions to humans. Simultaneously, the decrease of the HIR correspondingly results in a deterioration of human-agent interaction performance.

Human Simulation Due to the high cost of hiring annotators to label real human-agent collaboration data, it is costly for us to collect human-agent collaboration data on more datasets and, as a result, validate the efficacy of our method in broader scenarios. We instead use GPT-4 (gpt-4-0613) to build a simulation environment and make it collaborate with agents to solve tasks. This setup enables us to collect more "human-agent" collaboration data at a reasonable cost.

To verify the feasibility of using GPT-4 to simulate humans to collect "human-agent" collaboration data, we learn ReHAC on the HotpotQA GPT-4-agent collaboration data, named as ReHAC$_{\text{GPT-4}}$ and test its performance in the real human-agent collaboration environment. From Figure 2(a), we can see that ReHAC$_{\text{GPT-4}}$ exhibits better performance compared to ReHAC$_{\text{Human}}$ in human-agent collaboration when $\lambda = 0.06$ and 0.08. We suppose that this is possibly attributed to individual differences among humans, leading to a distribution variance in the human-agent collaboration data, while GPT-4-agent collaboration data exhibits higher consistency and lower variance. This makes ReHAC$_{\text{GPT-4}}$ learn the collaboration signal more easily, and thus is more stable and performs better.

To further reduce costs and observe the reward variation of ReHAC during the training process, we use GPT-4 to simulate humans in the evaluation phase. Figure 2(b) shows the evaluation results when using GPT-4 to simulate humans for collaboration. Comparing the results in Figure 2(a) and (b), we notice that the relative performance of various methods is generally consistent in both human-agent collaboration and GPT-4-agent collaboration. For example, the rewards R of ReHAC consistently surpass those of the Prompt method, and both ReHAC and the Prompt method outperform the Random method. This demonstrates the viability of using GPT-4 to simulate humans for evaluation.

Considering feasibility and cost-effectiveness, we will continue to use GPT-4 as a substitute for human participants in all subsequent extension experiments.

Learning Curves Figure 3 shows the learning curves during the training process. The curves are obtained by assessing the policy model's rewards

---

[1] The formula for calculating the human intervention rate is in Appendix A.3.

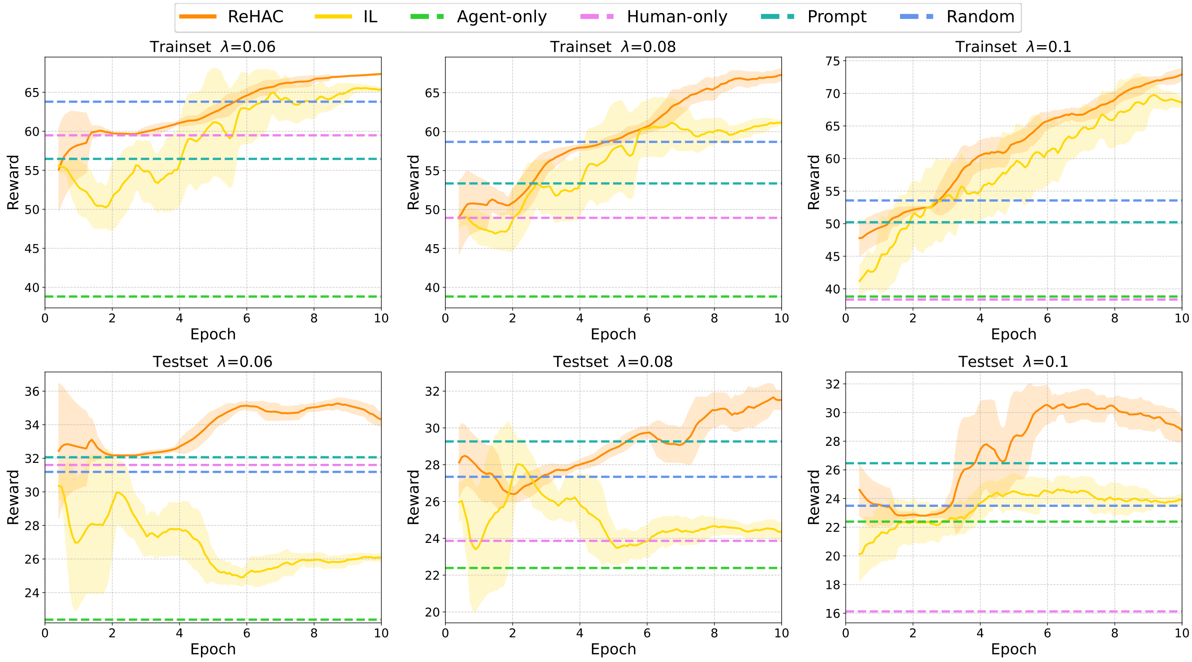


Figure 3: Reward R variations of different methods during the training process on HotpotQA dataset. Here we set the human intervention penalty coefficient λ to 0.06, 0.08, and 0.1. Curves of ReHAC and IL are averaged over 15 points, with shadows indicating the variance.

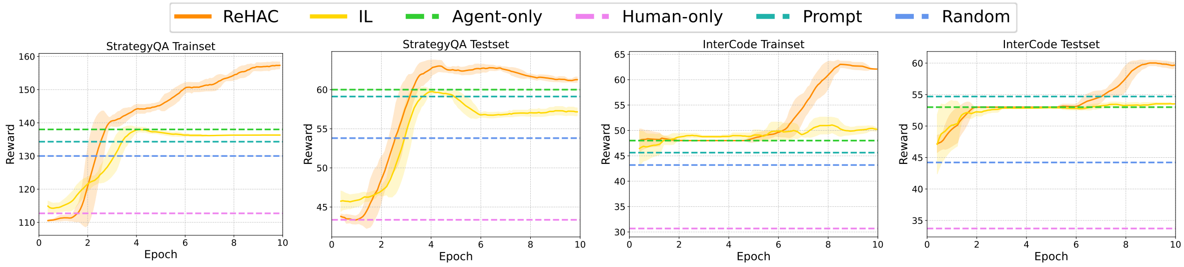


Figure 4: Reward R variations during the training process on three datasets. Curves of ReHAC and IL are averaged over 15 points, with shadows indicating the variance.

on the trainset and testset every 5 steps. From the figure, we can observe that (1) the rewards of ReHAC gradually increase during the training process, indicating that ReHAC can progressively identify suitable points to introduce human interventions. (2) While the IL method achieves high rewards on the trainset, it performs poorly on the testset. This suggests our RL-based learning method learns a more generalized human-agent collaboration strategy compared to directly learning the optimal strategy with the imitation learning method.

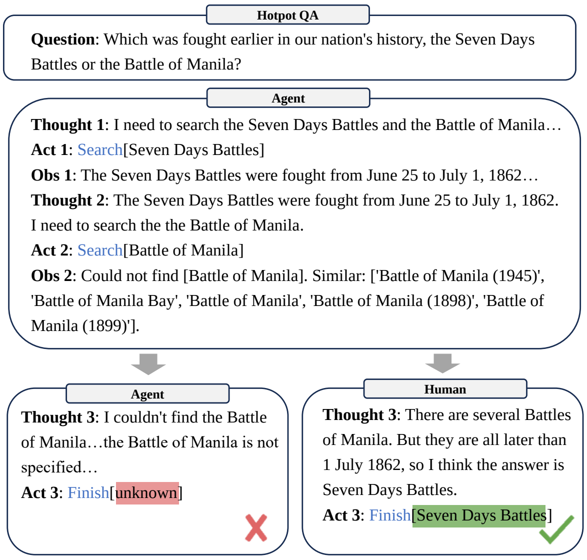
3.3 Performance on Different Dataset

In this part, we train and test ReHAC method on StrategyQA, and InterCode datasets in the GPT-4 simulation environment. For all experiments, we fix the parameter $\lambda = 0.08$. Throughout the training phase, we evaluate the policy model's rewards on the trainset and testset every 5 steps. Experimental results are shown in Figure 4. From the figure, we observe that: (1) Our proposed ReHAC method achieves higher reward scores compared to other baselines on all datasets. This validates the effectiveness of our approach across a broader range of datasets. (2) Both ReHAC and IL exhibit low variance and stability during the training process. Although our method and the IL method show a continuous reward increase during the training process, ReHAC can ultimately achieve higher rewards compared to the IL method. This indicates that our reinforcement learning-based method can provide more valuable guidance to the policy model $\pi_{\theta_1}^{collab}$, enabling it to determine when to introduce human interventions and consequently achieving higher rewards.

In summary, our method demonstrates superior

Dataset	Model	HIR (%)	Task Reward T	Reward R
HotpotQA	LLaMA-7B	51.46	46.90	31.38
	LLaMA-13B	47.64	46.78	32.22
InterCode	LLaMA-7B	4.15	62.00	60.08
	LLaMA-13B	3.10	60.00	58.56

Table 1: Experimental results regarding different model scales. HIR represents the human intervention rate.



Question: Which was fought earlier in our nation's history, the Seven Days Battles or the Battle of Manila?

Thought 1: I need to search the Seven Days Battles and the Battle of Manila…
Act 1: Search[Seven Days Battles]
Obs 1: The Seven Days Battles were fought from June 25 to July 1, 1862…
Thought 2: The Seven Days Battles were fought from June 25 to July 1, 1862. I need to search the the Battle of Manila.
Act 2: Search[Battle of Manila]
Obs 2: Could not find [Battle of Manila]. Similar: ['Battle of Manila (1945)', 'Battle of Manila Bay', 'Battle of Manila', 'Battle of Manila (1898)', 'Battle of Manila (1899)'].

Thought 3: I couldn't find the Battle of Manila…the Battle of Manila is not specified…
Act 3: Finish[unknown]

Thought 3: There are several Battles of Manila. But they are all later than 1 July 1862, so I think the answer is Seven Days Battles.
Act 3: Finish[Seven Days Battles]

Figure 5: Case Study. When the agent completes the task, the third step cannot be answered due to the ambiguity of the problem identified; using our method, the first two simple retrieval tasks are assigned to the agent to complete, while the third step is assigned to humans. Humans can complete the correct answer through bold speculation

performance across all datasets, affirming its ability to achieve an optimal balance between efficiency and effectiveness.

3.4 Scaling Analysis of Policy Model

In this section, we analyze the impact of the model scale on the performance of the policy model. Here, we set $\lambda = 0.08$ and conduct experiments on HotpotQA and InterCode datasets. As shown in Table 1, the LLaMA-7B model performs competitively with the LLaMA-13B model. This suggests that the Llama2-7B model is already proficient in handling the human-agent collaboration task, and the benefit of increasing the size of the model is slight. We will explore smaller policy model size in the future.

3.5 Case Study

In this part, we give a specific case on the HotpotQA dataset, as illustrated in Figure 5, to show how human-agent collaboration helps the complex task-solving. The task is to determine which historical event, the Seven Days Battles or the Battle of Manila, occurred first. When given the entire problem, the agent accurately determines the date of the Seven Days Battles but encounters multiple entries for the Battle of Manila, resulting in ambiguity. Consequently, the agent deems the query ambiguous and opts to respond with “unknown”. On the contrary, our ReHAC method requires the human intervention in this situation. Upon examining the related entries, the human observes that all mentioned dates for the Battle of Manila occurs after to July 1, 1862. Based on this insight, he conjectures that the Seven Days Battles occurred first. Although this conjecture is not absolutely certain, it represents the most likely decision based on the available information. Thus, our ReHAC method returns a correct response “Seven Days Battles”. This case also highlights an insightful aspect of our research into LLM-based agents: Researchers are committed to eliminating hallucinations in large language models (LLMs) to create rigorous and accurate intelligent agents. However, many tasks require imagination and intuition, making it crucial to integrate human creative thinking through human-agent collaboration at this juncture.

4 Discussion

In this paper, we conduct a preliminary exploration of key aspects of human-agent collaboration, aiming to lay the groundwork for further research in this field. Despite progress, unresolved problems and potential challenges persist. We propose three extended research directions to enhance the effectiveness, safety, and intelligence of human-agent collaboration:

Multi-level Human-Agent Collaboration Our focus is on modes where humans directly replace agents in action. However, given the distinct advantages of both humans and agents, we see a need to explore more complex collaboration levels. This includes human involvement in feedback, decision modification, and planning.

Development Stages of LLM-based Agents Inspired by the L1 to L5 grading model in autonomous driving, we suggest adapting this framework for LLM-based human-agent collaboration. It offers a clear structure to assess the current development stage of human-agent technologies and guide future research. While LLM agents have not reached high or full automation, this framework is crucial for identifying key technologies and challenges. However, our research indicates a significant gap before LLM agents achieve full automation (L5). Effective human-agent collaboration could be a bridge towards this goal.

Safety and Super Alignment Safety is paramount in human-agent collaboration, particularly in high-risk scenarios. It's vital to explore methods to secure the collaboration process and mitigate risks. Moreover, with the potential of LLM-based agents evolving into superintelligence, effective collaboration becomes increasingly crucial. This collaboration is key, as it not only allows humans to guide ethical and safety decisions but also ensures the alignment of LLM-based agents' objectives with human interests.

5 Related Work

LLM-based Agent Recent advancements in LLMs have demonstrated their capabilities in reasoning (Wei et al., 2022; Kojima et al., 2022; Hao et al., 2023; Luong et al., 2024; Yue et al., 2023) and task planning (Yao et al., 2023a; Kong et al., 2023; Shen et al., 2023; Yao et al., 2023b; Deng et al., 2023). These capabilities lay the foundation for the development of LLM-based agents (Shridhar et al., 2021; Yang et al., 2023; Liu et al., 2023b; Song et al., 2023; Wang et al., 2023a). LLM-based agents, which can interact with the environment and select subsequent actions based on environment feedback, have been applied in many domains, including web navigation (Nakano et al., 2021; Cheng et al., 2024; He et al., 2024), software engineering (Qian et al., 2023; Hong et al., 2023), and robotics (Wang et al., 2024; Mahadevan et al., 2024). By synergizing the reasoning and action abilities of LLMs, ReAct (Yao et al., 2022) incorporates environment feedback into reasoning traces and determines the next step action dynamically. Subsequent research focuses on integrating code (Wang et al., 2023b; Roziere et al., 2023; Xu et al., 2023), memory modules (Rana et al., 2023; Park et al., 2023), experience reflection (Shinn et al., 2023; Zhao et al., 2023a), and tools into LLM-based agents (Liu et al., 2023a; Patil et al., 2023; Qin et al., 2023), thereby augmenting their abilities in solving complex problems. However, current LLM-based agents still perform poorly on some complex tasks. This work aims to introduce human interventions and enable humans and agents to collaboratively address complex tasks, thereby achieving improved task performance.

Human-Agent Collaboration In Human-Agent Collaboration (HAC), traditional research has been centered on improving the naturalness and efficiency of human interactions with intelligent agents like robots and AI systems, effectively meeting human needs (Wang et al., 2021; Wu et al., 2022b). The rise of large-scale language models (LLM-based agents) marks a significant shift in the field, underscoring the role of human feedback and reasoning in enhancing agent capabilities. This approach leverages human insights to refine performance and decision-making processes. Recent studies employ heuristic rules to direct these agents towards seeking human assistance (Cai et al., 2023; Wu et al., 2022a; Mehta et al., 2023). Furthermore, there is an increasing emphasis on developing specialized prompts that motivate LLM-based agents to proactively seek human input, thus nurturing a more interactive and collaborative dynamic in these partnerships (Huang et al., 2022; Wang et al., 2023b). However, the effectiveness of these methods relies on designing high-quality rules or prompts. This is highly dependent on the designer's domain knowledge. Poor design may result in a system that cannot accurately understand or respond to complex task requirements. Our research focuses on designing a generalised and learnable method that coordinates human to effectively work with LLM-based agents in the form of direct planning.

6 Conclusion

In this paper, we propose the problem of large language model-based human-agent collaboration, delving into the synergy of human intuition and expertise with the computational prowess of LLM-based agents, particularly emphasizing their application in intricate decision-making tasks. We introduce a reinforcement learning-based approach for human-agent collaboration, named ReHAC. Central to ReHAC is a learnable policy model designed to pinpoint the most critical junctures for human intervention within the task-solving trajectory. Our experimental results show that ReHAC aspects better results and is more generalizable than heuristic rule-based or prompt-based approaches in human-agent collaboration tasks. We believe that ReHAC offers a practical pathway for the application of llm-agents in real-world scenarios.

Ethical Considerations and Limitations

The objective of this work focuses on human-agent collaboration, which requires humans to interact with LLM-based agents. We acknowledge that agents are likely to output some hallucinations and misleading information, and it is unclear how these contents impact humans. Additionally, all datasets used in this work are publicly available, and therefore, there are no data privacy concerns. All data collected will be used for research purposes only

The limitations of this paper can be summarised in three aspects:

1) The current study is confined to basic LLM-based agent architectures based on the "ReAct" and "Try Again" frameworks, while more complex architectures involving self-reflection and memory capabilities are still unexplored.

2) Our research primarily focuses on the use of 7B and 13B scale models as policy models for task allocation. Future work will investigate the feasibility of smaller models in carrying out these tasks, aiming to maintain performance while reducing resource consumption.

3) This study is based on the assumption that human performance supersedes that of agents. However, as technology advances, agents might surpass human capabilities. Future research will thus shift towards exploring human-agent collaboration models in this new context. Emphasis will be placed on assessing how human-agent collaboration can ensure the safety of agent decisions while aligning with human preferences.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13.

Ashish Bastola, Julian Brinkley, Hao Wang, and Abolfazl Razi. 2024. Driving towards inclusion: Revisiting in-vehicle interaction in autonomous vehicles. *arXiv preprint arXiv:2401.14571*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zefan Cai, Baobao Chang, and Wenjuan Han. 2023. Human-in-the-loop through chain-of-thought. *arXiv preprint arXiv:2306.07932*.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. 2024. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 902–909.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. 2023. Plug-and-play policy planner for large language model powered dialogue agents. *arXiv preprint arXiv:2311.00262*.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. 2024. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 910–919.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Conference on Empirical Methods in Natural Language Processing*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

M. A. Khan et al. 2022. Level-5 autonomous driving—are we there yet? a review of research literature. *ACM Computing Surveys*, 55(2):Article 27.

Julia Kiseleva, Alexey Skrynnik, Artem Zholus, Shrestha Mohanty, Negar Arabzadeh, Marc-Alexandre Côté, Mohammad Aliannejadi, Milagro Teruel, Ziming Li, Mikhail Burtsev, Maartje ter Hoeve, Zoya Volovikova, Aleksandr Panov, Yuxuan Sun, Kavya Srinet, Arthur Szlam, and Ahmed Awadallah. 2022. Iglu 2022: Interactive grounded language understanding in a collaborative environment at neurips 2022.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Yilun Kong, Jingqing Ruan, Yihong Chen, Bin Zhang, Tianpeng Bao, Shiwei Shi, Guoqing Du, Xiaoru Hu, Hangyu Mao, Ziyue Li, et al. 2023. Tptu-v2: Boosting task planning and tool usage of large language model-based agents in real-world systems. *arXiv preprint arXiv:2311.11315*.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023a. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023b. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*.

Karthik Mahadevan, Jonathan Chien, Noah Brown, Zhuo Xu, Carolina Parada, Fei Xia, Andy Zeng, Leila Takayama, and Dorsa Sadigh. 2024. Generative expressive robot behaviors using large language models. *arXiv preprint arXiv:2401.14673*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Nikhil Mehta, Milagro Teruel, Patricio Figueroa Sanz, Xin Deng, Ahmed Hassan Awadallah, and Julia Kiseleva. 2023. Improving grounded language understanding in a collaborative environment by interacting with agents through help feedback. *arXiv preprint arXiv:2304.10750*.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*.

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

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *7th Annual Conference on Robot Learning*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

SAE International. 2021. Sae levels of driving automation refined for clarity and international audience. `https://www.sae.org/news/2021/05/sae-j3016-driving-automation-levels`.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI tasks with chatGPT and its friends in hugging face. In *Advances in Neural Information Processing Systems*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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

Chao Wang, Stephan Hasler, Daniel Tanneberg, Felix Ocker, Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, and Michael Gienger. 2024. Large language models for multi-modal human-robot interaction. *arXiv preprint arXiv:2401.15174*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023a. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2023b. Mint: Evaluating llms in multi-turn interaction with tools and language feedback.

Zijie J Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. *arXiv preprint arXiv:2103.04044*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022a. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–22.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022b. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

Yiheng Xu, Hongjin Su, Chen Xing, Boyu Mi, Qian Liu, Weijia Shi, Binyuan Hui, Fan Zhou, Yitao Liu, Tianbao Xie, et al. 2023. Lemur: Harmonizing natural language and code for language agents. *arXiv preprint arXiv:2310.06830*.

John Yang, Akshara Prabhakar, Karthik R Narasimhan, and Shunyu Yao. 2023. Intercode: Standardizing and benchmarking interactive coding with execution feedback. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. 2023b. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Olaf Zawacki-Richter, Victoria I Marín, Melissa Bond, and Franziska Gouverneur. 2019. Systematic review of research on artificial intelligence applications in higher education–where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1):1–27.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023a. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023b. A survey of large language models.

A Appendix

A.1 Experimental Details

Model Implementation In our most experiments, we use Llama-2-7b-hf[2] downloaded from Huggingface as our policy model $\pi_{\theta_1}^{collab}$. We also conduct experiments based on Llama-2-13b-hf[3] model (see Section 3.3). We implement LoRA based on PEFT (Mangrulkar et al. (2022)) and set $r_{\text{LoRA}} = 16$ and $\alpha_{\text{LoRA}} = 16$ for all experiments. Based on Yao et al. (2022) and Yang et al. (2023), we set the step threshold for HotpotQA, StrategyQA, and InterCode to 7, 5, and 8, respectively. All experiments are conducted on NVIDIA A100 GPUs with 40GB memory.

Human-Agent Dataset For a real human-agent collaboration dataset, we employ a uniform sampling method where each action a_t has a 50% probability of being assigned to either a human annotator or the ChatGPT. For each question, we sample as many interaction trajectories as possible. Specifically, for each time t, we aim to sample trajectories including $a_t^{collab} = 0$ and $a_t^{collab} = 1$. Considering the diversity of responses from different annotators, we permit repeated sampling of the same trajectory during uniform sampling, which means all a_t^{collab} of two trajectories are the same. To enhance the quality of annotation, annotators are allowed to reference GPT-4's answers. We recruit 14 annotators through social media, all of whom are graduate students with strong language and reasoning skills. They are asked to annotate a total of about 2000 trajectories in four days and they get paid about $10 an hour. They were explicitly told that the data would be used to train the model and made public and that all the labeled data was unrelated to any individual's privacy. To facilitate the annotation process, we develop a graphical user interface (GUI)[4] and provide one hour of training to annotators. The collected data details are in Table 2.

GPT-4-Agent Dataset For the dataset constructed using GPT-4 to simulate human annotation, we adopt the same sampling method as human-agent dataset collection. However, due to the uniform or near-uniform distribution of GPT-4's responses, we skip duplicate paths during uniform sampling. Collected data details are listed in Table 2.

A.2 Baselines Details

Random We randomly choose a human or an agent to conduct action a_t at a probability of 50%.

Prompt We prompt an agent to actively decide action a_t should be finished by itself or a human. The related prompts are shown in Table 5 and Table 6. Experimental results of Random and Prompt are averaged over three repeated experiments.

Imitation Learning We select the top 50% of actions that receive the highest rewards in each state s_t as expert demonstrations. These expert demonstrations (state-action pairs) are then used to supervise the fine-tuning of the policy model. This approach allows the policy model to learn how to make decisions that get a higher return in a given state.

Dataset	Trainset		Testset
	Questions	Trajectories	Questions
HotpotQA(real)	141	1937	100
HotpotQA(sim)	141	2135	100
StrategyQA(sim)	250	2420	100
InterCode(sim)	100	2071	100

Table 2: Collected dataset details. Questions mean the number of questions we used for human-agent collaboration task. Trajectories mean the overall trajectory number we collected. (real) refers to the real human-agent collaboration dataset, and (sim) refers to the human-agent collaboration dataset collected by using GPT-4 to simulate humans.

A.3 Human Intervention Rate

We denote the number of steps completed by humans and agents in the dataset by num_h and num_a, respectively. The Human Intervention Ratio (HIR) is calculated as

$$\text{HIR} = \frac{num_h}{num_h + num_a}.$$

HIR measures the rate of human intervention. Generally, a higher HIR indicates better task performance, but it also tends to increase costs.

---

[2] https://huggingface.co/meta-llama/Llama-2-7b-hf

[3] https://huggingface.co/meta-llama/Llama-2-13b-hf

[4] The GUI is as shown in Figure 6.

Experiment	α	ϵ	Learning Rate	Batch Size
HotpotQA$_{\lambda=0.06}$(GPT-4-agent, 7b)	0	0.3	3e-5	64
HotpotQA$_{\lambda=0.08}$(GPT-4-agent, 7b)	0		3e-5	
HotpotQA$_{\lambda=0.10}$(GPT-4-agent, 7b)	0		5e-5	
HotpotQA$_{\lambda=0.08}$(GPT-4-agent, 13b)	0.1		3e-5	
HotpotQA$_{\lambda=0.06}$(human-agent, 7b)	0.05		5e-5	
HotpotQA$_{\lambda=0.08}$(human-agent, 7b)	0.1		5e-5	
HotpotQA$_{\lambda=0.1}$(human-agent, 7b)	0.0		5e-5	
StrategyQA$_{\lambda=0.08}$(GPT-4-agent, 7b)	0.1		1e-5	
InterCode$_{\lambda=0.08}$(GPT-4-agent, 7b)	0		5e-5	
InterCode$_{\lambda=0.08}$(GPT-4-agent, 13b)	0.05		5e-5	

Table 3: Hyper-parameter settings for all experiments.

Methods	HotpotQA			StrategyQA			InterCode		
	HIR (%)	Task Reward	Reward	HIR (%)	Task Reward	Reward	HIR (%)	Task Reward	Reward
Agent-only	0.0	22.39	22.39	0.0	60.00	60.00	0.0	53.00	53.00
Human-only	100.0	54.82	23.86	100.0	68.00	43.36	100.0	73.00	33.72
Random	50.84	42.73	27.34	49.50	65.67	53.8	50.09	66.00	44.21
Prompt	34.06	40.46	29.26	9.14	61.33	59.12	9.94	59.33	54.69
IL	22.08	31.50	24.70	4.76	59.00	57.88	1.01	54.00	53.52
Ours	51.46	46.90	**31.38**	20.47	66.00	**61.12**	4.15	62.00	**60.08**

Table 4: ReHAC$_{\text{GPT-4}}$ Human intervention rate (HIR), task reward T, and reward R of different methods on GPT-4-agent testsets.

Imagine you are a clever planner.

Given an unfinished trajectory with several steps, your task is to decide whether the next step should be carried out by ChatGPT or a human. This decision should be based on a thoughtful evaluation of the difficulty of the next step and the progress made in the current trajectory. Here are two finished trajectory examples.
Example 1:
${example1}
Example 2:
${example2}
Now please decide whether the next step should be carried out by ChatGPT or a human. Please consider the following factors:
1. If the next step is relatively straightforward and well within ChatGPT's capabilities, instruct ChatGPT to proceed with the next step. If the task is deemed challenging or requires human judgment, recommend human intervention.
2. If the trajectory has been consistently handled by ChatGPT without notable issues, encourage ChatGPT to continue. If there have been challenges or uncertainties in the trajectory, consider suggesting human involvement for the next step.
3. Note that human intervention will significantly increase the cost, so try to balance the accuracy and efficiency.
If the next step should be carried out by ChatGPT, return [ChatGPT], otherwise, return [Human]. Only return [ChatGPT] or [Human].

#Your unfinished trajectory#: ${current trajectory}
#Your return#:

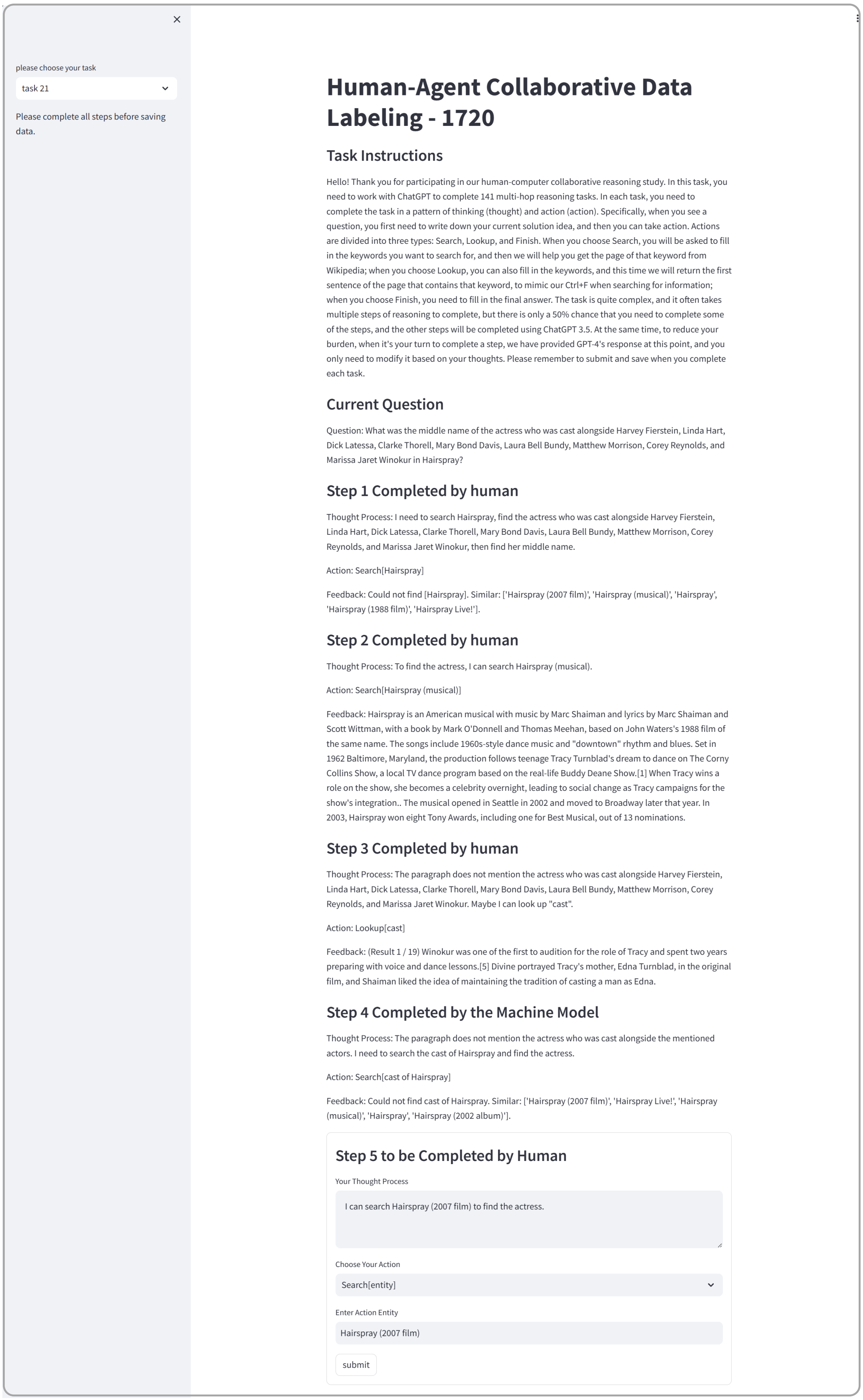
Table 5: The prompt template used for the prompt-based method in QA dataset.

Imagine you are a clever planner in SQL.

Given an unfinished trajectory with several SQL commands, your task is to decide whether the next command should be carried out by ChatGPT or a human. This decision should be based on a thoughtful evaluation of the difficulty of the next command and the progress made in the current trajectory. Here are two finished trajectory examples.
Example 1:
${example1}
Example 2:
${example2}
Now please decide whether the next command should be carried out by ChatGPT or a human. Please consider the following factors:
1. If the next command is relatively straightforward and well within ChatGPT's capabilities, instruct ChatGPT to proceed with the next command. If the task is deemed challenging or requires human judgment, recommend human intervention.
2. If the trajectory has been consistently handled by ChatGPT without notable issues, encourage ChatGPT to continue. If there have been challenges or uncertainties in the trajectory, consider suggesting human involvement for the next command.
3. Note that human intervention will significantly increase the cost, so try to balance the accuracy and efficiency.
If the next command should be carried out by ChatGPT, return [ChatGPT], otherwise, return [Human].
Only return [ChatGPT] or [Human].

#Your unfinished trajectory#: ${current trajectory}
#Your return#:

Table 6: The prompt template used for the prompt-based method in InterCode dataset.



please choose your task: task 21

Please complete all steps before saving data.

Human-Agent Collaborative Data Labeling - 1720

Task Instructions

Hello! Thank you for participating in our human-computer collaborative reasoning study. In this task, you need to work with ChatGPT to complete 141 multi-hop reasoning tasks. In each task, you need to complete the task in a pattern of thinking (thought) and action (action). Specifically, when you see a question, you first need to write down your current solution idea, and then you can take action. Actions are divided into three types: Search, Lookup, and Finish. When you choose Search, you will be asked to fill in the keywords you want to search for, and then we will help you get the page of that keyword from Wikipedia; when you choose Lookup, you can also fill in the keywords, and this time we will return the first sentence of the page that contains that keyword, to mimic our Ctrl+F when searching for information; when you choose Finish, you need to fill in the final answer. The task is quite complex, and it often takes multiple steps of reasoning to complete, but there is only a 50% chance that you need to complete some of the steps, and the other steps will be completed using ChatGPT 3.5. At the same time, to reduce your burden, when it's your turn to complete a step, we have provided GPT-4's response at this point, and you only need to modify it based on your thoughts. Please remember to submit and save when you complete each task.

Current Question

Question: What was the middle name of the actress who was cast alongside Harvey Fierstein, Linda Hart, Dick Latessa, Clarke Thorell, Mary Bond Davis, Laura Bell Bundy, Matthew Morrison, Corey Reynolds, and Marissa Jaret Winokur in Hairspray?

Step 1 Completed by human

Thought Process: I need to search Hairspray, find the actress who was cast alongside Harvey Fierstein, Linda Hart, Dick Latessa, Clarke Thorell, Mary Bond Davis, Laura Bell Bundy, Matthew Morrison, Corey Reynolds, and Marissa Jaret Winokur, then find her middle name.

Action: Search[Hairspray]

Feedback: Could not find [Hairspray]. Similar: ['Hairspray (2007 film)', 'Hairspray (musical)', 'Hairspray', 'Hairspray (1988 film)', 'Hairspray Live!'].

Step 2 Completed by human

Thought Process: To find the actress, I can search Hairspray (musical).

Action: Search[Hairspray (musical)]

Feedback: Hairspray is an American musical with music by Marc Shaiman and lyrics by Marc Shaiman and Scott Wittman, with a book by Mark O'Donnell and Thomas Meehan, based on John Waters's 1988 film of the same name. The songs include 1960s-style dance music and "downtown" rhythm and blues. Set in 1962 Baltimore, Maryland, the production follows teenage Tracy Turnblad's dream to dance on The Corny Collins Show, a local TV dance program based on the real-life Buddy Deane Show.[1] When Tracy wins a role on the show, she becomes a celebrity overnight, leading to social change as Tracy campaigns for the show's integration.. The musical opened in Seattle in 2002 and moved to Broadway later that year. In 2003, Hairspray won eight Tony Awards, including one for Best Musical, out of 13 nominations.

Step 3 Completed by human

Thought Process: The paragraph does not mention the actress who was cast alongside Harvey Fierstein, Linda Hart, Dick Latessa, Clarke Thorell, Mary Bond Davis, Laura Bell Bundy, Matthew Morrison, Corey Reynolds, and Marissa Jaret Winokur. Maybe I can look up "cast".

Action: Lookup[cast]

Feedback: (Result 1 / 19) Winokur was one of the first to audition for the role of Tracy and spent two years preparing with voice and dance lessons.[5] Divine portrayed Tracy's mother, Edna Turnblad, in the original film, and Shaiman liked the idea of maintaining the tradition of casting a man as Edna.

Step 4 Completed by the Machine Model

Thought Process: The paragraph does not mention the actress who was cast alongside the mentioned actors. I need to search the cast of Hairspray and find the actress.

Action: Search[cast of Hairspray]

Feedback: Could not find cast of Hairspray. Similar: ['Hairspray (2007 film)', 'Hairspray Live!', 'Hairspray (musical)', 'Hairspray', 'Hairspray (2002 album)'].

Step 5 to be Completed by Human

Your Thought Process: I can search Hairspray (2007 film) to find the actress.

Choose Your Action: Search[entity]

Enter Action Entity: Hairspray (2007 film)

Figure 6: Human-Agent collaborative labelling user interface