A Lightweight Reasoning Method with Test-Time Scaling for Preserving Diversity and Factuality in LLM-Based Decision-Making

Rongrong Chen
Peking University
Beijing, China
chenrongrong@stu.pku.edu.cn

Kailin Gao ✉*
AI Lab, Lenovo Research
Beijing, China
gaokl6@lenovo.com

Yuan He
AI Lab, Lenovo Research
Beijing, China
heyuan1@lenovo.com

Hongsheng Qi
AI Lab, Lenovo Research
Beijing, China
qihs1@lenovo.com

Abstract

Large Language Models (LLMs) have shown remarkable performance across various tasks, but their reasoning capabilities still face challenges. This paper aims to mitigate the limitations of LLMs in complex decision-making tasks, which require high-level reasoning ability. We introduce *Smart Peers*, a lightweight reasoning method designed to enhance LLMs' performance in decision-making tasks by integrating test-time scaling. Specifically, *Smart Peers* employs sequential and parallel self revision to perform task decomposition, enabling the LLM to make independent decisions multiple times and has the opportunity to revise its decision based on all peers' decisions. In this case, the method achieves test-time scaling, thereby ensuring diversity and factuality at each step of the decision-making task and enhancing the overall task completion. As a lightweight method, *Smart Peers* demonstrates superior performance compared to other complex trajectory planning algorithms in certain tasks in our experiments. We evaluate *Smart Peers* on three decision-making tasks: WebShop, ALFWorld, and Mini-Crosswords. The results demonstrate that *Smart Peers* achieves significant performance improvements over baseline methods. In particular, on the WebShop task, *Smart Peers* achieves a relative improvement of approximately 34.63% compared to other baseline methods. Additionally, *Smart Peers* exhibits notable advantages, including fully leveraging the LLMs' capability and promptly correcting erroneous steps, laying a foundation for future research in complex reasoning.

CCS Concepts

• **Computing methodologies → Artificial intelligence**.

*Corresponding author: Kailin Gao (gaokl6@lenovo.com)

KDD 2025 Workshop on Inference Optimization for Generative AI, Toronto, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

Large Language Models, Decision Making, Collaborative Reasoning, Test-Time Scaling, Task Decomposition

ACM Reference Format:
Rongrong Chen, Kailin Gao ✉, Yuan He, and Hongsheng Qi. 2025. A Lightweight Reasoning Method with Test-Time Scaling for Preserving Diversity and Factuality in LLM-Based Decision-Making. In *Proceedings of (KDD 2025 Workshop on Inference Optimization for Generative AI).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Large Language Models (LLMs) excel in many tasks [1] and are applied to decision-making tasks that require high-level reasoning abilities [9, 11, 23, 24], yet their reasoning faces challenges [17].

Studies address the challenge along two paths. (i) Task decomposition. e.g., the Chain-of-Thought (CoT) family [19, 20, 22, 23], generates intermediate steps yet under-uses LLMs, yielding unreliable results. For example, ReAct in the WebShop task [21] had both successful and failed trajectories (see Appendix C). (ii) Experience learning. e.g., Reflexion [11], yet it also has limitations in tough tasks. Reflexion shows no improvement on the WebShop task due to the task's highly diverse search space and need for high-precision search queries, which Reflexion struggles to handle effectivel[11].

To address these limitations, we retain the idea of task decomposition, consider exploring more diverse generations at each decomposition step. Since LLMs can generate both successful and failed attempts for a step in different trials. We suggest a mechanism where, when the LLM makes a wrong decision, we pause and let it refer to the trajectories it could potentially generate, deciding whether to revise the current decision. As alternative outputs from the LLM might have made the correct decision, if the decision-making LLM opts to follow the correct action, the trajectory that would have failed could be transformed into a successful one. A detailed example analysis is provided in Appendix C.

Based on these ideas, we propose *Smart Peers*. We treat the LLM's independent task reruns as peers. *Smart Peers* enables the LLM to draw on the decisions of other peers for performing better task decomposition, thereby accomplishing the decision-making task. Specifically, the key point of *Smart Peers* is that at each step in the task decomposition process, the LLM is allowed to make its own decision while also having the opportunity to revise its decision based on the reasoning of other peers.

We have implemented *Smart Peers* in various decision-making tasks, where *Smart Peers* demonstrates its specific strengths. In particular, on the WebShop task, *Smart Peers* achieves a relative improvement of approximately 34.63% compared to other baseline methods. Moreover, further analysis indicates that *Smart Peers* contributes to fully leveraging the capabilities of LLMs.

2 Related Works

The CoT Family. CoT prompting [20] enables complex reasoning in LLMs with few examples. ReAct [23] adds decision-making through integrated reasoning and action, while Tree of Thought (ToT) [22] explores multiple reasoning paths via a tree structure.

Multi-Model Synergy for Task Solving. Inspired by human collaboration, recent studies [4, 6] explore how multiple LLMs can improve performance through interaction or debate, highlighting the potential of multi-model cooperation in problem-solving.

Intermediate Step Revision. Revising intermediate steps is key for decomposable tasks, enabling iterative refinement of solutions. THOUGHTSCULPT [3] leverages Monte Carlo Tree Search (MCTS) to guide LLMs in continuous self-revision, improving output quality without altering model architecture.

Test-Time Scaling (TTS). TTS enhances LLMs' inference by adjusting computational resources, mainly through Verifier-Based Search[5, 7, 18] and Refining the Proposal Distribution [8, 10]. TTS offers a cost-effective alternative to model expansion, often outperforming larger models by optimizing test-time computation. It also supports LLMs' progressive self-improvement by iteratively refining outputs and using verifiers to assess quality [14].

However, existing methods often underutilize LLMs or are overly complex. To address this, we propose Smart Peers, a more lightweight framework that effectively stimulates LLMs' latent reasoning capabilities.

3 Smart Peers

We propose *Smart Peers*, a reasoning method integrating sequential and parallel self revision inspired by human thinking behaviors. Next, we will introduce how this method works.

Assume a benchmark LLM L_0 that, in each task decomposition round, generates a primary decision with multiple alternatives, represented as a peer set $\{P_1, P_2, ..., P_n\}$. We evaluate the final completion of the task based on L_0's performance. $P_i (1 \leq i \leq n)$ are auxiliary peers. They assist L_0 in deciding whether to change the current action during the task execution.

For task T, the benchmark LLM L_0 and its peers will perform think-action-observation for R rounds to complete the task.[1] In round $r (1 \leq r \leq R)$, each of them will perform a thought (t_r^i), take an action (a_r^i) based on the current trajectory ($traj_r$), and obtain the observation (o_r^i). $i = 0$ denotes the benchmark LLM L_0, while indices $1 \leq i \leq n$ correspond to peers $P_i (1 \leq i \leq n)$.

Making Decision Independently. (See (a) in Figure 1) To better complete the task, we aim to explore more possibilities of trajectories, so we hope that LLM can generate more diverse thoughts and actions in each round. Therefore, in round r, L_0 generates the next thought and action multiple times, solely based on the basic prompt and the current trajectory. One of these generations serves as L_0's own decision while the others act as L_0's peers. These generations do not influence each other.

Deciding whether to Change. (See (b) in Figure 1) In round r, after L_0 and its peers have generated their thoughts and actions, we can obtain the corresponding observations. Next, we provide the T-A-O of the auxiliary peers to the benchmark LLM L_0. We then let L_0 decide whether to change its action based on its own T-A-O and that of its peers, i.e., in round r, L_0's decision is determined by $\{(t_r^i, a_r^i, o_r^i)\}_{0 \leq i \leq n}$. This process can be formalized as $a_r = S(\{(t_r^i, a_r^i, o_r^i)\}_{0 \leq i \leq n})$, where $S(\cdot)$ is the choosing process. a_r is selected from $\{a_r^i\}_{0 \leq i \leq n}$, that is, L_0 can choose to keep its own action or copy one of its peers' actions. In this step, L_0 does not blindly copy the actions of its peers but instead considers its own and its peers' T-A-O to decide whether to change its action.

Aligning. (See (c) in Figure 1) After L_0 makes its choice, we obtain the final t_r, a_r and o_r for the round r, and add them to $traj_r$, which formulates $traj_{r+1}$. In the next round $r+1$, the L_0 will continue to generate thoughts and actions based on $traj_{r+1}$.

Performing Making Decision Independently, Deciding whether to Change, and Aligning in each round, and finally, the completion of the task is determined by L_0's performance in the last round. [2]

Smart Peers integrates sequential and parallel reasoning. Specifically, the LLM engages in multiple T-A-O rounds, representing sequential reasoning. In each round, it first generates its own initial decision, then parallely generates other peers' decisions and evaluates them. Based on this evaluation, the LLM decides whether to revise its initial action. This process continues within each round, enabling the LLM to iteratively optimize its decisions.

While this iterative mechanism might resemble conventional multi-agent collaboration, they fundamentally differ in both methodology and objectives. Multi-agent frameworks focus on collaborative decision-making through debate [4] or group discussion [15], risking loss of agent independence and convergence to a potentially erroneous consensus. In contrast, *Smart Peers* enables peers to independently generate thoughts and actions and the benchmark LLM adjusts its action based on the T-A-O of all peers, emphasizing reasoning space expansion and diversity exploration.

Besides, *Smart Peers* stands out for its lightweight design. Unlike THOUGHTSCULPT, which uses computationally intensive Monte Carlo search, *Smart Peers* adopts a resource-efficient strategy, making it more practical when computational power is limited.

4 Experiment

4.1 Experiment Setup

Tasks and Datasets. We evaluate *Smart Peers*' performance on three decision-making tasks: WebShop [21], ALFWorld [13], Mini-Crosswords (scraped data from GooBix). The summary introduction is in Table 1; detailed descriptions and task-related experiment setups are in Appendix B and D.

Additionally, since "Deciding whether to Change" is a critical step in *Smart Peers*, we specifically describe the type of "T-A-O" for each task, as shown in Table 2.

[1] In the following, we will refer to "think-action-observation" as "T-A-O" as a shorthand.

[2] If the task is completed before round R, it will stop early.

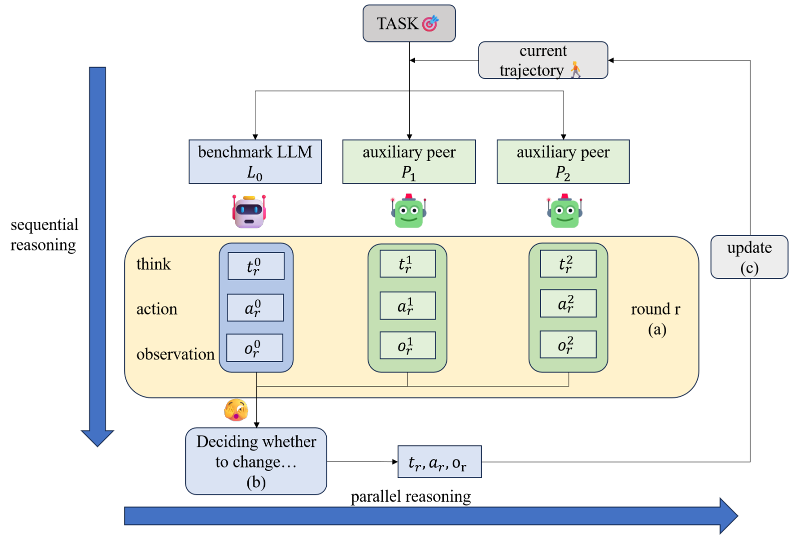

Figure 1: An intuitive illustration of *Smart Peers*

Table 1: The summary introduction of three decision-making tasks: WebShop, ALFWorld, Mini-Crosswords

Dataset	Goal	Example
WebShop [21]	Navigate through web interactions to purchase a product that matches a given user instruction.	I need a long clip-in hair extension which is natural looking, and price lower than 40.00 dollars.
ALFWorld [13]	Navigate and interact with a simulated household through text commands to complete a task.	You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 2, a desk 1, ... Your task is to: look at bowl under the desklamp.
Mini-Crosswords	Fill a 5x5 grid with letters to solve the crosswords.	HINT: Row 1: An agendum; something to be done/Row 2: An engine/.../Col 1: To heap/Col 2: An Indian antelope/...

Table 2: Type of "T-A-O" in each task

Dataset	Think&Action	Observation
WebShop	think[...], search[...], click[...]...	Current Page Content/SystemInfo (e.g., Invalid action!)
ALFWorld	think[...], go to..., pick...	SystemInfo (You open the drawer 2. ...In it, you see nothing.)
Mini-Crosswords	fill actions	Current board&Current evaluation metrics

Baselines. We compare *Smart Peers* with several widely used baseline methods. Specifically, WebShop and ALFWorld are compared with ReAct [23]. Mini-Crosswords, on the other hand, is compared with both ReAct and ToT [22]. To ensure fair comparison, *Smart Peers* uses the same few-shot exemplars and parameters as [23] for WebShop and ALFWorld, and those from [22] for Mini-Crosswords. Details are in Appendix D.

Evaluation. For WebShop, we use the average "score" (avg.score) as the evaluation metric, which is defined in [21] and reflects attribute matching between the purchased and desired item. For ALFWorld, we use the task completion rate (success rate) as the evaluation metric. For Mini-Crosswords, we use the average letter correctness rate (r_letter) and word correctness rate (r_word) of the final results as the evaluation metrics.

Implementation Details. We use qwen1.5-72b-chat for evaluating *Smart Peers* and baseline methods in main experiments, with 2 auxiliary peers for *Smart Peers* by default. Prompts and hyperparameter are detailed in Appendix D. All results are averaged over three runs. Besides, to systematically evaluate the performance gains of *Smart Peers* over ReAct across varying model scales, we extend our experiments to include different parameter sizes of the Qwen1.5 architecture. Specifically, while maintaining the baseline configuration of qwen1.5-72b-chat, we additionally evaluate both the smaller qwen1.5-7b-chat and the larger qwen1.5-110b-chat variants.

4.2 Results and Analysis

We report the results of *Smart Peers* on three decision-making tasks. Tabel 3 presents the comparison results with other baseline methods. The best results for each task are highlighted in bold. *Smart Peers* achieves notable performance improvements across these tasks. In the ALFWorld task, *Smart Peers* slightly surpasses ReAct in success rate. For the Mini-Crosswords task, *Smart Peers* outperforms ReAct and ToT. Most notably, in the WebShop task, *Smart Peers* significantly outperforms ReAct with the relative improvement of 34.63%, highlighting its superior performance in more complex reasoning tasks. Overall, these results demonstrate that *Smart Peers* exhibits better adaptability and effectiveness in handling diverse decision-making tasks, especially in complex reasoning scenarios.

Since *Smart Peers* achieves significant enhancements, we conduct further analysis. We first analyze how *Smart Peers* works. Secondly, we explore which types of tasks *Smart Peers* is suitable for. Finally, we experiment with the influence of model parameters on WebShop, specifically examining how the gains of *Smart Peers* compared to ReAct vary under different model parameters.

Table 3: Comparison on three decision-making tasks using *Smart Peers* and baselines

Datasets	Method	Metrics	Result*100
WebShop	ReAct	avg.score	39.36
	Smart Peers		**52.99**
ALFWorld	ReAct	success rate	66.17
	Smart Peers		**67.91**
Mini-Crosswords	ReAct	*r_letter*	29.20
		r_word	11.33
	ToT	*r_letter*	22.80
		r_word	9.67
	Smart Peers	*r_letter*	**32.67**
		r_word	**12.33**

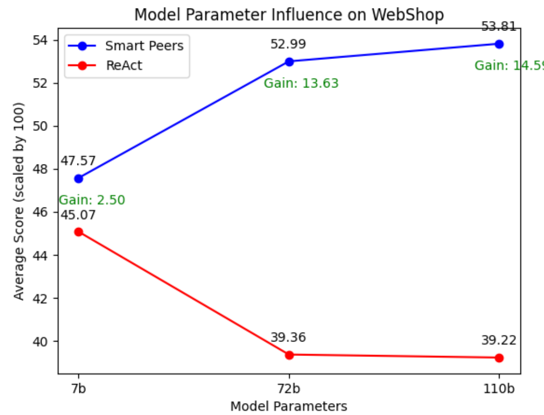

Figure 2: Model Parameter Influence on WebShop

How does *Smart Peers* work? From the experimental results, we find that *Smart Peers* successfully transform failure trajectories into successful ones. In the case shown in Appendix E, the benchmark LLM maintains its decision in the first three rounds, then chooses to "click[Sapphire Blue+Purple]" in the fourth. Since the true clickable button is [sapphire blue+purple], this action isn't recognized by the system. Meanwhile, an auxiliary peer directly chooses "click[Buy Now]" and completes the task. In this round, the benchmark LLM follows the auxiliary peer's decision, corrects its previous mistake, and succeeds. This case shows that *Smart Peers* ensures decision-making diversity at each step by allowing the benchmark LLM to act independently, enabling timely correction.

What tasks is *Smart Peers* suitable for? From Table 3, we can observe that *Smart Peers* shows significant improvements on WebShop and Mini-Crosswords; comparatively, the improvement on ALFWorld is smaller. The above observation indicates that *Smart Peers* is more suitable for tasks with "non-absolute metrics". We consider metrics that are either 0 or 1 to be "absolute metrics", such as in ALFWorld, where the evaluation is binary (completed or failed), with no intermediate states. This evaluation can be understood as having no "process points"; even if there is improvement during the process, the final result may still be a failure. However, in WebShop and Mini-Crosswords have a metrics on a 0-1 scale, focusing on task completion processes, offering a larger learning space, which is why significant improvements are observed in these tasks.

What are the gains of *Smart Peers* compared to ReAct under different model parameters? Given the advantages demonstrated by *Smart Peers* in certain decision-making tasks, we further explore whether this framework better leverages large-parameter LLMs. We employ LLMs with varying model sizes to compare the performance of *Smart Peers* and ReAct, in order to investigate whether this framework can fully utilize the capabilities of large-parameter LLMs compared to the baseline method. Specifically, we conduct experiments on the WebShop, using qwen1.5-7b-chat, qwen1.5-72b-chat, and qwen1.5-110b-chat to run both ReAct and *Smart Peers*, respectively. Results are shown in Figure 2.

From Figure 2, we observe an intriguing phenomenon: on WebShop, as model parameters increase, ReAct's performance gradually declines, while *Smart Peers* gradually improves. We hypothesize that larger model parameters generally enhance model capability, but in a specific task, larger model parameters may lead to "overthinking". [2] also mentions LLMs' "overthinking" phenomenon. [2] analyzes the performance of o1-like models on mathematical problems, pointing out that the o1 model exhibits significant "overthinking" on simple problems, generates many useless solutions and leads to a decline in model performance. In the context of WebShop, more precise interaction with the system is required. "overthinking" may lead to model missing the correct purchase timing (just like the failed trajectory shown in Appendix C). *Smart Peers*, by allowing the benchmark LLM to make independent decisions, mitigates this issue to some extent.

Additionally, as the model parameters increase, the gain of *Smart Peers* over ReAct also increases, showing a trend proportional to the model parameters increase. Specifically, from 7b to 72b, the parameters increase by approximately 10 times leads to a gain increase of about 5 times; from 72b to 110b, a parameter increase of about 1.5 times results in a gain increase of about 1.1 times.

From the above analysis, it can be seen that *Smart Peers* is capable of fully leveraging the capabilities of large-parameter LLMs, i.e., achieving better performance under larger parameter sizes.

5 Conclusion

In this paper, we introduce *Smart Peers*, a lightweight reasoning method with test-time scaling for LLM-Based decision-making. We believe that as the capabilities of LLMs continue to strengthen, how to fully leverage the potential of LLMs is a question we need to consider. *Smart Peers* achieves satisfactory results in decision-making tasks by integrating sequential and parallel self revision to perform task decomposition. Specifically, *Smart Peers* allows the benchmark LLM to make independent decisions as peers, and lets the benchmark LLM decide whether to change the current decision according to its peers. This paradigm enhances the diversity and factuality of reasoning. We conduct evaluations on three decision-making tasks, and the results demonstrate that our method outperforms some previous solutions. We hope that this work can serve as a foundation for further research, providing new perspectives on complex reasoning.

References

[1] 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* (2023).

[2] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do NOT Think That Much for 2+ 3=? On the Overthinking of o1-Like LLMs. *arXiv preprint arXiv:2412.21187* (2024).

[3] Yizhou Chi, Kevin Yang, and Dan Klein. 2024. THOUGHTSCULPT: Reasoning with Intermediate Revision and Search. *arXiv preprint arXiv:2404.05966* (2024).

[4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325* (2023).

[5] Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179* (2023).

[6] Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142* (2023).

[7] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

[8] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36 (2023), 46534–46594.

[9] OpenAI [n. d.]. https://openai.com/index/learning-to-reason-with-llms/. [Accessed 23-12-2024].

[10] Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching foundation model agents how to self-improve. In *Automated Reinforcement Learning: Exploring Meta-Learning, AutoML, and LLMs*.

[11] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[12] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

[13] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768* (2020).

[14] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314* (2024).

[15] Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280* (2023).

[16] Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuanjing Huang. 2022. BERTScore is unfair: On social bias in language model-based metrics for text generation. *arXiv preprint arXiv:2210.07626* (2022).

[17] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261* (2022).

[18] Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935* (2023).

[19] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

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

[22] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[23] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

[24] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19632–19642.

A Limitations and Broader Impacts

The experimental results indicate that *Smart Peers*, using relatively lightweight methods, can fully leverage the capabilities of LLMs compared to other baseline methods. At the same time, it ensures diversity in each step of decision-making and has the ability to promptly correct intermediate steps that may lead to failure, which helps mitigate the inherent biases and misconceptions [16] of a single model.

However, we think the method has the following limitations:

- **Instability.** In the *Smart Peers* framework, since both the benchmark LLM and peers utilize the same base LLM, they inherently possess equivalent capabilities. Consequently, their generations and self-revision process are essentially randomized events. For example, during our experiments, we found that sometimes the benchmark LLM does not choose the correct actions of other peers but insists that its own actions are correct (indeed, its action may be wrong), thereby introducing instability to the framework. To address this limitation, future implementations may incorporate capability differentiation among peers, such as employing a more advanced model as the benchmark LLM to undertake crucial self-revision tasks. This enhanced framework preserves the key of test-time scaling in *Smart Peers* while integrating multi-agent collaboration, presenting a promising direction for subsequent research exploration.
- **Cost increasing.** Although *Smart Peers* demonstrates certain advantages compared to the ReAct, the cost of utilizing LLMs and the time required to complete tasks are also increasing. Specifically, the cost is primarily driven by two factors: parallel generation of multiple independent decisions and sequential reasoning with lengthening reasoning trajectories, which will not only mean that each round of reasoning becomes more time-consuming, but also lead to a higher number of tokens being used, this further escalates the financial burden. Therefore, it is essential to consider the balance between task completion and cost.
- **Lack of prior planning.** *Smart Peers* relies on the inherent capabilities of LLMs to accomplish tasks. Consequently, there is an upper limit to its performance improvement. We can consider enhancing its effectiveness by introducing prior planning to refine the decision-making process. Specifically, we could allow the benchmark LLM to perform more informed self-revisions based on the pre-planned strategies of other peers, thereby strengthening the overall framework.

Based on the above limitations, our future research needs to consider better strategies based on *Smart Peers*, which includes integrating other advanced frameworks or developing more effective test-time scaling methods to further improve the decision-making capabilities of LLMs.

B Details of Datasets

The summary of datasets' information can be found in Table 1. The more detailed information of each dataset is shown as follows.

WebShop

WebShop is a recently introduced online shopping environment [21] designed to simulate real-world interactions in a noisy language setting. It features a vast database of 1.18 million real-world products and 12,000 human-generated instructions, making it a challenging platform for evaluating agents in practical applications. WebShop presents a high diversity of both structured and unstructured texts, such as product titles, descriptions, and options scraped from Amazon. The task requires an agent to navigate through web interactions to purchase a product that matches a given user instruction, for example, finding a nightstand with specific attributes like a nickel finish and a price under $140. The agent must perform actions such as searching for relevant items, selecting product options, and making a purchase based on the instruction. The performance of the agent is measured by the average score, which reflects the percentage of desired attributes covered by the chosen product across all episodes, and the success rate, indicating the percentage of episodes where the chosen product meets all user requirements.

In our experiments, we select 100 test instructions and use the average score as the primary evaluation metric. We compare *Smart Peers* with ReAct [23] in this dataset.

ALFWorld

ALFWorld [13] is a synthetic text-based game environment designed to mimic the challenges of the embodied ALFRED benchmark [12]. It features six types of tasks where an agent must achieve high-level goals, such as examining a paper under a desk lamp, by navigating and interacting with a simulated household through text commands. Each task instance can involve over 50 locations, thus demanding the agent to plan, track subgoals, and systematically explore the environment.

In our experiments, we evaluate on 134 evaluation games used in [13]. We use the prompts from ReAct [23], which constructs prompts for each task type using permutations of annotated trajectories. We use the success rate as the primary evaluation metric. We compare *Smart Peers* with ReAct [23] in this dataset.

Mini-Crosswords

The Mini-Crosswords dataset, sourced from GooBix, consists of 156 games of 5×5 puzzles. Each task provides 5 horizontal and 5 vertical clues, and the goal is to fill a 5×5 grid with letters to solve the crosswords. The evaluation measures success at three levels: the percentage of correct individual letters, complete words, and solved games.

In our experiments, we use the setup from ToT [22], i.e., to ensure diversity and avoid similar clues between adjacent games, we select 20 games with indices 1, 6, ..., 91, 96 for testing, and we use the percentage of correct individual letters and complete words as main evaluation metrics. We compare *Smart Peers* with ReAct [23] and ToT [22] in this dataset.

C the ReAct Performance in WebShop Task

As an example, Table 4 illustrates an example of using ReAct to complete the WebShop task [21], which involves interacting with a shopping website to purchase desired items. The score indicates the degree of task completion, with a higher score indicating better task performance. Multiple attempts using the ReAct method resulted in both successful and failed trajectories. It is evident that such methods are not stable, as the task can sometimes be completed successfully and other times fail. This indicates that these methods have not yet fully and stably leveraged the capabilities of LLMs.

In the example, the failed trajectory resulted from "overthinking". Indeed, [B08NSH1ZN4] and [B082WZFD19] are both products that meet the requirements. However, in the failed attempt, the LLM did not directly choose to purchase the product. Instead, it opted to check the product's attributes and reviews. After reviewing this information, the LLM decided to buy the product. Unfortunately, the LLM had forgotten that the "[Buy Now]" button was not on the current page. As a result, its action was non-interactive with the system, leading to the failure, while the alternative outputs generated by the LLM are likely to have made the correct decision, i.e., directly purchasing the correct product, which shows the intermediate revision is feasible, i.e.,we could pause after making the wrong decision, allowing the current decision-making LLM to refer to the trajectories it could potentially generate and decide whether to revise the current decision. If the current decision-making LLM decides to follow the action that has made the correct decision, the trajectory that should have failed can be changed to success.

D Implementation Details

Experiment Setup. In our experiment, for each task, the number of tasks used, the evaluation metrics and the comparison with other methods can be seen in Table 5.

LLMs Settings and Prompts. We access the qwen1.5-7b-chat, qwen1.5-72b-chat, and qwen1.5-110b-chat models through API. Among these, the qwen1.5-72b-chat model is primarily used, while qwen1.5-7b-chat and qwen1.5-110b-chat models are employed for comparative experiments. Following the settings of previous works, the temperature during the generation process is set to 0 for WebShop and ALFWorld (consistent with [23]), and to 0.7 for Mini-Crosswords (consistent with [3][3]).

Regarding the prompts, we maintain consistency with previous studies in terms of instruction, few-shot examples, and so forth. Specifically, for the ReAct method, we use the prompts from the code released by [23][4]. For the ToT method, we use the prompts from the code released by [3][5]. For *Smart Peers*, since it is based on the ReAct method in the "Making Decision Independently" step, the prompts used here are consistent with those of ReAct. The prompts used in the "Deciding whether to Change" step are illustrated in the Table 6.

Hyperparameter Settings. The hyperparameters for LLMs have been previously mentioned. Here, we specify the hyperparameter settings for the number of execution rounds. For WebShop, both ReAct and *Smart Peers* execute 15 rounds (i.e., 15 cycles of think-action-observation); for ALFWorld, both ReAct and *Smart Peers* execute 20 rounds; and for Mini-Crosswords, ReAct and *Smart Peers* execute 20 rounds, while ToT executes 60 rounds (i.e., the

[3] In [3], the proposed method is compared with ToT.
[4] https://github.com/ysymyth/ReAct
[5] https://github.com/cyzus/thoughtsculpt

Table 4: Example of Successful and Failed Trajectories in WebShop. Blue text (Action, Observation) represents the framework of ReAct. Bold text (search, think, click, etc.) indicates the types of Action. Yellow background signifies current clickable buttons. Red text "Invalid action!" indicates that the Action type cannot interact with the system. In the example, clicking a button that does not exist on the current page leads to "Invalid action!".

<table>
<tr><th>Successful Trajectory</th><th>Failed Trajectory</th></tr>
<tr><td colspan="2">Instruction: I need a long clip-in hair extension which is natural looking, and price lower than 40.00 dollars</td></tr>
<tr><td colspan="2">Action: **search**[long clip-in hair extension natural looking price < 40.00]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B08NSH1ZN4] #Some descriptions#...$3.99
[B08883YHLS] #Some descriptions#...$100.0
[B082WZFD19] #Some descriptions #...$9.99</td></tr>
<tr><td>Action: **think**[B08NSH1ZN4 is a long clip-in hair extension in the desired color and at an affordable price. ....]
Observation: OK.
Action: **click**[B08NSH1ZN4]
Observation:
[Back to Search]
[< Prev]
size [16 inch (pack of 1)] [17 inch]
color [1pcs] [brown&black] [purple& black]
Price: $3.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]
Action: **click**[Buy Now]
Observation: Your score (min 0.0, max 1.0): 1.0</td><td>Action: **think**[B08NSH1ZN4 and B082WZFD19 are both long clip-in hair extensions with prices lower than 40.00 dollars. B082WZFD19 seems more suitable as it specifically mentions being natural looking.]
Observation: OK.
Action: **click**[B082WZFD19]
Observation:
[Back to Search]
[< Prev]
color [blue] [purple] [purple+ princess pink]
Price: $9.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]
Action: **think**[...I should check its attributes before purchasing.]
Observation: OK.
Action: **click**[Attributes]
Observation:
[Back to Search]
[< Prev]
hair extensions
easy apply
...
Action: **think**[...I'll proceed to read reviews to ensure it meets the requirement.]
Observation: OK.
Action: **click**[Reviews]
Observation: Invalid action!
Action: **click**[Buy Now]
Observation: Invalid action!
...
Observation: Your score (min 0.0, max 1.0): 0.0</td></tr>
</table>

depth of the tree is 60), using the best result from the last round as the final result for ToT.

E Case Study

Table 7 shows a case which indicates that *Smart Peers* successfully changed the failed trajectory into the successful trajectory.

Table 5: Experimental setup

Dataset	Number of tasks used	Evaluation metrics used	Comparison with other methods
WebShop	100 test instructions	Average score (reflects the percentage of desired attributes covered by the chosen product)	ReAct [23]
ALFWorld	134 evaluation games in [13]	Success rate (whether the task is completed)	ReAct [23]
Mini-Crosswords	20	The percentage of correct individual letter, complete words	ReAct[23], ToT [22]

Table 6: The prompt used for "Deciding whether to Change" with the number of auxiliary peers set to 2 as an example

Prompt for Deciding whether to Change
Now you are doing a task. Following is your so far trajectory. (SO FAR TRAJECTORY BEGIN) {so far trajectory} (SO FAR TRAJECTORY END) When deciding which action should do next, firstly you make an action and get the corresponding observation: YOUR ACTION:actions[0] YOUR OBSERVATION:observations[0] Now you'd like to check that whether your action is properly. Fortunately, you have two peers who are doing the same task with you. Their so far trajectories are same as you. And their follow action and observation are: PEER 1 ACTION:actions[1] PEER 1 OBSERVATION:observations[1] PEER 2 ACTION:actions[2] PEER 2 OBSERVATION:observations[2] Based on the next actions and corresponding observations made by you and your peers, you can consider to change or maintain your action, if you think one of your peers is doing better than you, you can change your action to your peers' actions. However, if you think you are doing better than others, you can choose to maintain your action. In summary, You have 3 choices: X. maintain your initial action. Y. follow PEER 1's action. Z. follow PEER 2's action. Please carefully consider which action is most advantageous. Please return the option directly, no explanation needed. For example, if you decide to choose X, then return 'X' directly. Now it's your turn! Please give your answer!

Table 7: A case study

Instruction:
I need a long clip-in hair extension which is natural looking, and price lower than 40.00 dollars.

[benchmark LLM]
Round 1:
Action: search[long clip-in hair extension natural looking price < 40.00]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B08NSH1ZN4]
Colored Hair Extensions,Colored Party Highlights Clip On In Hair Extensions,Multi-Colors Party Highlights Streak Synthetic Hairpieces Wonderful Rainbow ColorsPurple & Black,1pcs)
$3.99
[B08883YHLS]
18 Inch Long 140G Thick Corn Wave Hair Bun Extensions Kinky Curly Bun Heat Resistant Drawstring Ponytail Hairpiece Scrunchie Accessories for Women / 33#…
$100.0
[B082WZFD19]
Wiysday 20PCS Colored Clip in Hair Extensions Party Highlights Clip in Colored Hair Extensions for Kids Blue Hair Extensions Clip in 22 inch Straight Synthetic Hairpieces. (Sapphire Blue+Purple)
$9.99
=== Round 1 choose to maintain own decision ===
Round 2:
Action: think[B08NSH1ZN4 and B082WZFD19 seem to be within the budget but B08NSH1ZN4 is colored and might not be natural looking. B082WZFD19 is 22 inches long and has multiple colors which could potentially provide a more natural look when mixed with natural hair.]
Observation: OK.
=== Round 2 choose to maintain own decision ===
Round 3:
Action: click[B082WZFD19]
Observation:
[Back to Search]
[< Prev]
color [blue][purple][purple+ princess pink][rainbow colors][red][sapphire blue+purple][sapphire blue+teal blue]
Wiysday 20PCS Colored Clip in Hair Extensions Party Highlights Clip in Colored Hair Extensions for Kids Blue Hair Extensions Clip in 22 inch Straight Synthetic Hairpieces. (Sapphire Blue+Purple)
Price: $9.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]
=== Round 3 choose to maintain own decision ===
Round 4:
Action: click[Sapphire Blue+Purple]
Observation: Invalid action!
One of other auxiliary peers action and observation:
Action: click[Buy Now]
Observation: Your score (min 0.0, max 1.0): 1.0
=== Round 4 choose to follow other auxiliary peer's decision ===
Task is Solved!
