

FISRF: Flexible Instructional Scaffolding Reflection Framework for Robotics Task Planning with Large Language Model

Anonymous Author(s)

Abstract—Large Language Models (LLMs) are widely adopted for reflective error correction in task planning; however, existing approaches primarily rely on autonomous exploration, rendering them prone to accumulating erroneous experiences during early interactions with new environments. To address this limitation, we propose the Flexible Instructional Scaffolding Reflection Framework (FISRF), an agent–mentor architecture that estimates the robot’s Zone of Proximal Development (ZPD) in task planning and adaptively selects reflection strategies and mentor guidance, thereby facilitating the accumulation of effective error-correction experience. Experiments in ALFWorld and on real-world robotic platforms demonstrate that FISRF reduces the average number of error-correction rounds by 37% and redundant reasoning by 25.1%, while improving the overall task success rate by 8%. Project webpage (no author information):<https://fisrfplanning.github.io/>

I. INTRODUCTION

Recent advances in robot task planning increasingly leverage large language models (LLMs) to generate coherent multi-step plans [1], [2]. Despite strong performance on complex tasks, LLM-based planners remain susceptible to hallucinations and logical inconsistencies [3]. To alleviate these limitations, reflection mechanisms have been proposed. RETROFORMER [4] and Reflexion [5] extend the ReAct [6] framework by incorporating reflective feedback through textual gradients. Another research line [7], [8], [9] integrates tree-search strategies with reflection loops to support more deliberate exploration. More recent systems, such as Expel [10] and AutoManual [11], construct environment-specific rule libraries that enable structured retrospective error analysis. Furthermore, FCRF [12] and Thinkless [13] quantify task complexity to adaptively adjust reflection intensity (see Figure 1), thereby transforming reflection from a fixed procedure into a flexible mechanism.

Most existing reflection methods rely on robots operating autonomously under predefined reflection procedures, enabling effective experience accumulation in familiar or structured tasks. However, when robots enter new environments, they often lack effective experience to handle unfamiliar problems and may be unable to correct seemingly simple errors. These early-stage errors can accumulate, leading to numerous ineffective attempts and resulting in reduced efficiency and degraded performance. (Figure 1) We therefore ask: *How can robots rapidly accumulate effective experience in new environments while maintaining a certain degree of autonomy?*

To address these limitations, we build on the theory of **Instructional Scaffolding** from educational psychology [14]. This theory suggests that a **learner’s problem-solving ability improves most effectively when guidance is provided**

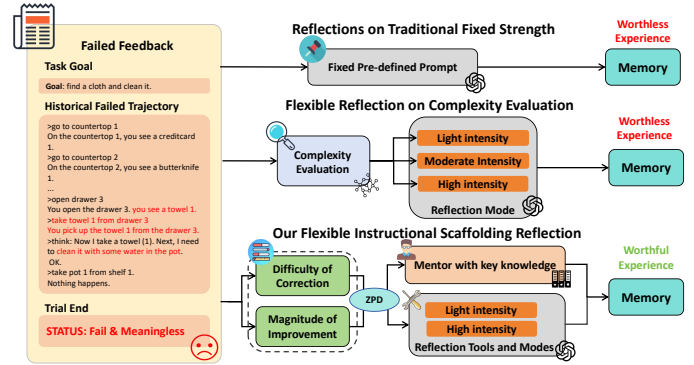


Fig. 1. Compared with existing approaches, FISRF adaptively determines whether to solicit mentor guidance and selects appropriate reflection strategies according to the difficulty of correction and the magnitude of improvement, in each trial, thereby effectively preserving and leveraging valuable experience for subsequent correction.

within their Zone of Proximal Development (ZPD)—the gap between what can be accomplished independently and what can be achieved with appropriate guidance. When persistent obstacles arise during independent exploration, targeted instruction that addresses core difficulties facilitates knowledge internalization and transfer, enabling more robust mastery of complex skills.

This raises a key challenge: **the lack of a quantitative and operational formulation of the robot’s Zone of Proximal Development in task planning**. Addressing this problem requires formalizing and automating the dynamic guidance mechanisms traditionally provided by human teachers. Specifically, the robot’s task performance must be quantitatively and adaptively modeled to establish a capability baseline and delineate its developmental boundaries. Based on this baseline, appropriate guidance strategies should be automatically generated. Moreover, the framework must progressively withdraw external support and consolidate assisted experience into autonomous, generalizable competence, thereby forming a closed loop from guided learning to independent execution.

Motivated by the aforementioned challenge, we propose the **Flexible Scaffolding Reflection Framework (FISRF)**, an online approach that provides adaptive guidance to large language models based on task performance. FISRF is grounded in instructional scaffolding theory and adopts an agent–mentor architecture. The **Agent module** consists of three components: 1) an **Actor** LLM that generates action sequences; 2) an **Evaluator** LLM that determines ZPD entry by dynamically modeling **two ZPD factors**: difficulty of correction (DC) and magnitude of improvement (MI); and 3) **Self-**

Reflection LLMs that perform experience summarization and lesson extraction. The **Mentor module** is implemented as a retrieval-augmented generation (RAG) component with a structured knowledge base. It retrieves task-relevant context and generates state-conditioned guidance grounded in the agent’s reflective feedback. After task execution, the agent evaluates its performance to determine ZPD entry. If the condition is satisfied, it requests mentor guidance and integrates the retrieved guidance with self-reflection for subsequent planning, thereby reducing redundant trials and enabling efficient, adaptive experience accumulation.

To assess the effectiveness of the proposed framework, we conduct experiments on the ALFWorld [15] household planning benchmark. The tasks in the dataset are systematically organized into six categories based on their operational characteristics. Empirical results indicate that our approach attains superior performance across all task types. In summary, our contributions are fourfold:

- 1) We propose a Flexible Instructional Scaffolding Reflection Framework (FISRF) that enables rapid accumulation of effective experience.
- 2) We formalize the zone of proximal development for robotic task planning using difficulty of correction and improvement magnitude.
- 3) We introduce an adaptive agent–mentor architecture that dynamically balances autonomous reflection and mentor guidance.
- 4) We validate FISRF through extensive experiments in ALFWorld and real-world robotic tasks, demonstrating significant improvements in planning efficiency and task success rates.

II. RELATED WORKS

A. LLMs for Task Planning

Several prior works have explored the use of LLM for task planning, motivated by their strong reasoning and long-horizon decision-making capabilities. For instance, LID [16] employs a pre-trained GPT-2 [17] as a unified framework for interactive decision processes, reformulating observations, goals, and historical trajectories into sequential representations that are encoded and passed to a policy network for action prediction. ReAct [6] integrates explicit reasoning with action generation, enabling iterative deliberation and execution. Building on this paradigm, methods such as CodeAsPolicy [18], ProgPrompt [1], and AdaPlanner [19] leverage code synthesis to represent plans as executable programs, thereby improving structural consistency and interpretability. In embodied robotics, representative work such as SayCan [20] grounds high-level linguistic reasoning in real-world affordances, constraining generated actions to ensure feasibility. Similarly, LLMPlanner [21], [22] extends LLM-based planning frameworks to embodied decision-making scenarios, further demonstrating their potential in robotic task planning.

B. LLM Agents for Self-Reflection

Building upon early one-shot planning strategies, subsequent research introduces self-reflection mechanisms that incorporate feedback for multi-step planning and iterative error correction in LLM-based agents. For instance, extending the ReAct framework, Reflexion [5] enables agents to analyze prior failures using environmental feedback and refine their plans in subsequent trials. Expel [10] further develops this paradigm by introducing an offline learning stage, where agents accumulate experience across tasks through trial-and-error interactions. AutoManual [11] constructs a structured representation of the environment to facilitate effective multi-task planning. Additionally, several approaches [7], [8], [9] integrate tree search with reflective mechanisms to improve solution quality during task execution. Building upon the fixed reflection intensity paradigm described above, we propose adaptive reflection strategies in FCRF [12] that are conditioned on task complexity, leading to improved reflection efficacy and overall planning performance.

III. PRELIMINARIES

A. Planning Framework

A task can be formally represented as a tuple $\langle G, S, O, A, T \rangle$, where G specifies the goal condition, S denotes the state space, O corresponds to the observation space, A is the action set, and $T : S \times A \rightarrow S$ defines the state transition dynamics induced by action execution.

The objective of planning is to identify a policy—typically instantiated as an action sequence—that guides the system from an initial state to a target state satisfying G . In LLM-based planning frameworks, such structured task elements are commonly encapsulated in a natural language description D provided to the model.

B. Self-Reflection Process in LLM-based Planning

Following Reflexion [5], we formalize Reflection as an iterative optimization procedure guided by environmental feedback. At each trial t , the planner interacts with the environment to generate a trajectory τ_t , and obtains a scalar evaluation $r_t = M_E(\tau_t)$. The reflection module maps the pair (τ_t, r_t) into a linguistic feedback signal $sr_t = M_{\text{ref}}(\tau_t, r_t)$, which is stored in memory mem and leveraged to update the policy in subsequent trials. The iterative loop (τ_t, r_t, sr_t) continues until the environment feedback indicates that the task has been successfully completed.

IV. METHODOLOGY

In this section, we describe the Flexible Instructional Scaffolding Reflection Framework (FISRF). As shown in Figure 2, FISRF adopts an actor–mentor architecture that consists of four parts: an LLM as planning Actor, the Flexible Self-Reflection Process, the short-long memory management and the overall adaptive Evaluator model. We provide a detailed description of each of these models and subsequently elucidate their collaborative functioning within the FISRF framework.

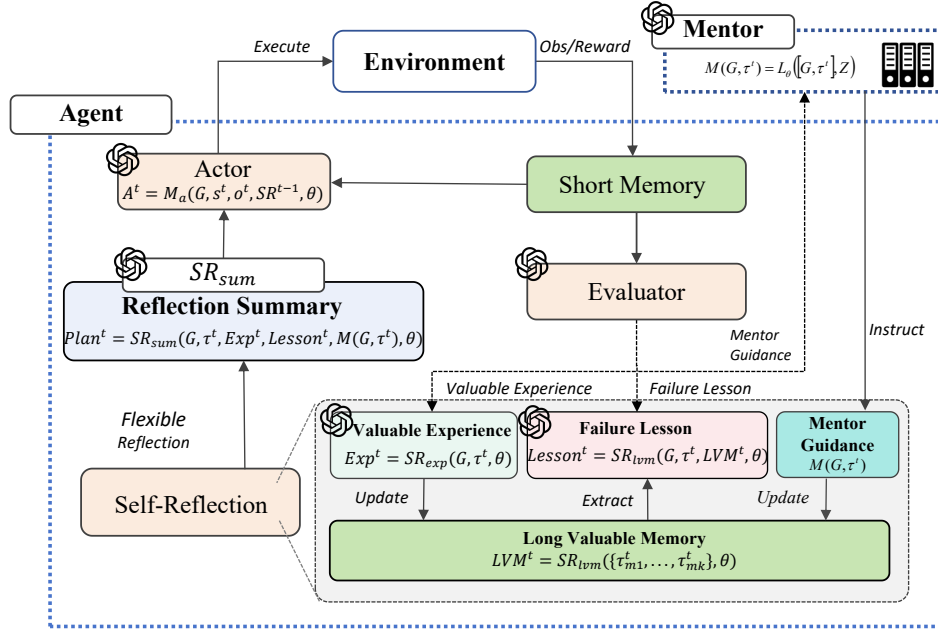


Fig. 2. The framework of FISRF. The planning process is executed by the Actor, the evaluation process is executed by Evaluator, the reflection process is executed by the Self-Reflection, the difficulty of correction and magnitude of improvement are assessed, according to the assessment results, determine whether to seek mentor guidance and implement appropriate reflection strategies. By integrating insights from autonomous exploration with mentor guidance, our framework inform subsequent plan. The resulting reflections and planning trajectories are stored in a memory module for long-term retrieval.

A. LLM as planning Actor

Adopting the architectural paradigm of the Reflexion framework, the planning procedure is executed by an LLM-based *Actor*, denoted as M_a . At each iteration, M_a processes the task goal, current environmental observations, and reflection summaries generated from previous failed trials. The model is prompted to generate a textual action sequence as output. The planning objective of M_a is formulated as:

$$A^t = M_a(G, s^t, o^t, SR^{t-1}, \theta), \quad (1)$$

where $A^t = \{a_1^t, \dots, a_i^t\}$ denotes the current action sequence, G represents the task goal, o^t represents the environmental observations at trial t , $SR^{t-1} = \{sr^{t-k}, \dots, sr^{t-1}\}$ denotes the set of self-reflection summaries from the previous k trials under consideration, and θ represents the parameters of M_a . The action sequence generated by the Actor LLM M_a is stored in the memory module *mem* to support the subsequent self-reflection process, and this iterative procedure continues until the current task is successfully completed.

B. The Short-Long Memory Management

During task planning, maintaining contextual memory is essential for the LLM. However, incorporating all historical memory into the prompt introduces redundancy, increasing cognitive burden and potentially degrading performance. Inspired by the long short-term memory mechanism in the Reflexion framework, we design a short-term memory module *sm* and a long-term valuable memory module *lvm* for structured memory management. The module *sm* serves as an external buffer, comprising a trajectory management component T^t and a reflection management component R^t , enabling systematic organization of task trajectories and

reflection content. Specifically, T^t stores the task trajectory τ^t of the current trial t . In contrast, R^t preserves all failure reflection records up to trial t , denoted as $\{r^1, \dots, r^t\}$, for selective retrieval by the self-reflection module in subsequent trials. The module *lvm* stores key knowledge provided by the mentor, as well as valuable experiences accumulated through autonomous exploration, serving as a universal knowledge base for the task environment. This hierarchical memory mechanism mitigates the cognitive burden on the LLM while enhancing memory organization efficiency and reflection effectiveness.

C. The RAG-Based Mentor Module

The mentor module, denoted as M , employs a Retrieval-Augmented Generation (RAG) architecture that integrates a parameterized language model L_θ with a scenario-specific knowledge base D . This hybrid approach enables the retrieval of task-relevant context to mitigate the limitations of static parametric knowledge. Given a task goal G and the current trajectory τ^t , the knowledge extraction and generation process is formulated as:

$$M(G, \tau^t) = L_\theta([G, \tau^t], Z) \quad (2)$$

where $Z = \text{Top-}k_{z \in D}(\text{sim}(E_\phi(G, \tau^t), E_\phi(z)))$ represents the set of the k most relevant documents retrieved from D . Here, E_ϕ denotes the embedding model used to compute cosine similarity between the query and the knowledge base entries. The retrieved evidence Z is concatenated with the query $[G, \tau^t]$ and fed into L_θ to produce grounded guidance.

D. The Self-Reflection Model

The Self-Reflection framework, denoted as *SR*, comprises three sub-modules: SR_{exp} , SR_{lvm} , and SR_{sum} . The re-

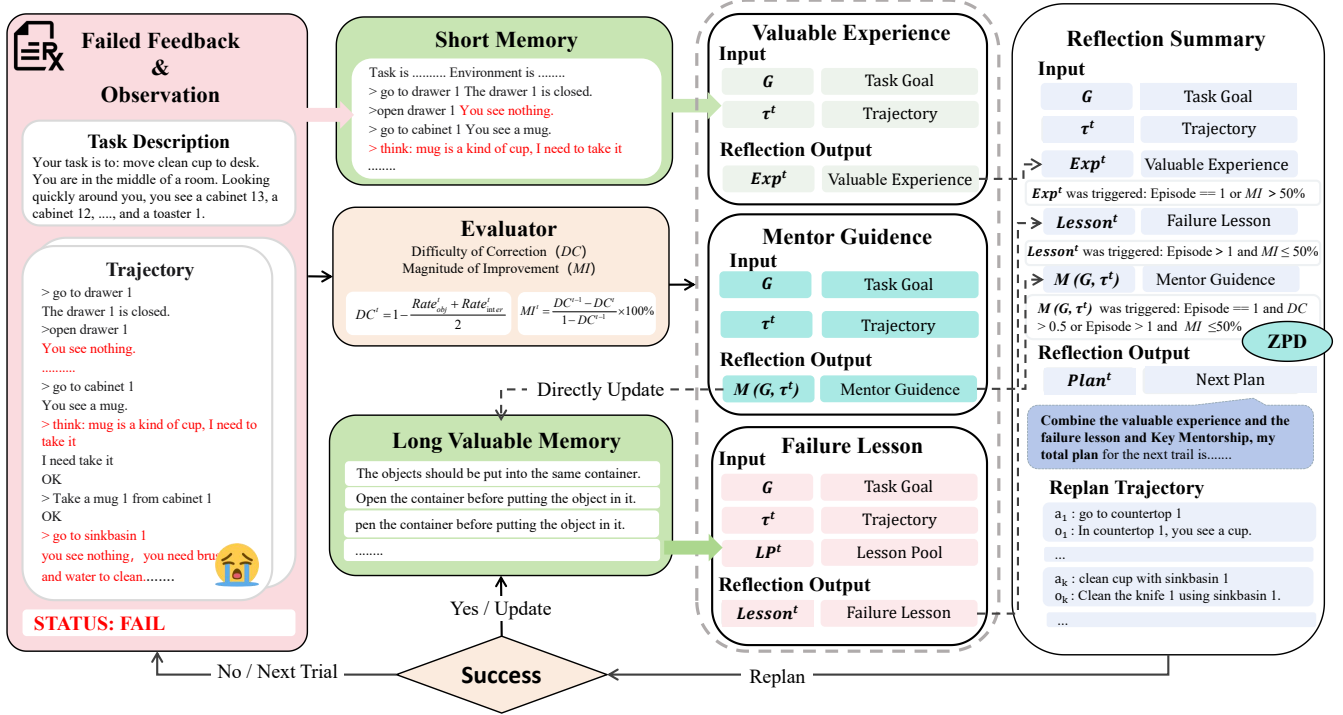


Fig. 3. The detailed illustration of FISRF. The difficulty of correction and magnitude of improvement are first assessed, determining reflection strategies and whether to seek mentor guidance. After that, the detailed self-reflection process consists of the valuable experience summary process, the failure lesson summary process, Seek mentor guidance and the Reflection Summary process. Final output of the self-reflection process is a improved new plan for the next attempt.

reflection process is executed in three stages: (i) distilling high-value experiences from short-term memory, (ii) extracting failure-based lessons from long-term memory, and (iii) synthesizing a comprehensive reflection summary integrated with mentor-provided knowledge. The resulting output is stored in the short-term memory module sm , enabling the actor model M_a to retrieve and leverage these insights for informed planning in subsequent trials.

a) *The valuable experience summary model:* This process is implemented in the submodule SR_{exp} . Specifically, SR_{exp} analyzes the trial trajectory generated by SR_a , identifies effective interaction attempts from unsuccessful trials, and distills them into reusable experience. The extracted insights are stored in memory to support subsequent reflection. The procedure can be summarized as follows:

$$Exp^t = SR_{exp}(G, \tau^t, \theta), \quad (3)$$

where G represents the task goal, τ^t denotes the trajectory of the current trial t , and θ represents the parameters of FSR_{exp} , which are primarily instantiated through prompt design.

b) *The failure lesson summary model:* This model is completed by the submodule SR_{lvm} and consists of two components: the preliminary maintenance of the long valuable memory and the extraction of failure lessons.

During long valuable memory maintenance, SR_{lvm} accesses existing cross-task trajectories online, summarizing universal failure lessons from all trajectories successfully corrected prior to the current episode, then incrementally

updates lvm with new lessons, formulated as:

$$LVM^t = SR_{lvm}(\{\tau_{m1}^t, \dots, \tau_{mk}^t\}, \theta), \quad (4)$$

where LVM^t represents the up-to-date long valuable memory set up to trail t , $\{\tau_{m1}^t, \dots, \tau_{mk}^t\}$ represents all k successfully modified task trajectories in the current trail t , θ represents the parameters of SR_{lvm} , mainly embodies through prompt.

During failure lesson extraction, SR_{lvm} analyzes the current task trajectory against the task goal to retrieve the most relevant failure lesson from M_l , serving as external context for reflection. This lesson, combined with valuable experience Exp^t , informs subsequent reflection content generation, formulated as:

$$Lesson^t = SR_{lvm}(G, \tau^t, LVM^t, \theta), \quad (5)$$

where G represents the task goal, τ^t represents the trajectory of current trail t , LP^t represents the up-to-date long valuable memory set up to trail t , θ represents the parameters of SR_{lvm} , mainly embodies through prompt.

Upon completion of the experience and failure lesson summarization processes, the reflection summary submodule SR_{sum} performs **reflection summarization**, consolidating successful and failed experiences with key mentor guidance $M(q)$ to generate an improved plan. The reflected new constructive plan will be stored in the sm module and guide the planning process of M_{actor} in the next trail. The comprehensive construction process can be described as:

$$Plan^t = SR_{sum}(G, \tau^t, Exp^t, Lesson^t, M(G, \tau^t), \theta) \quad (6)$$

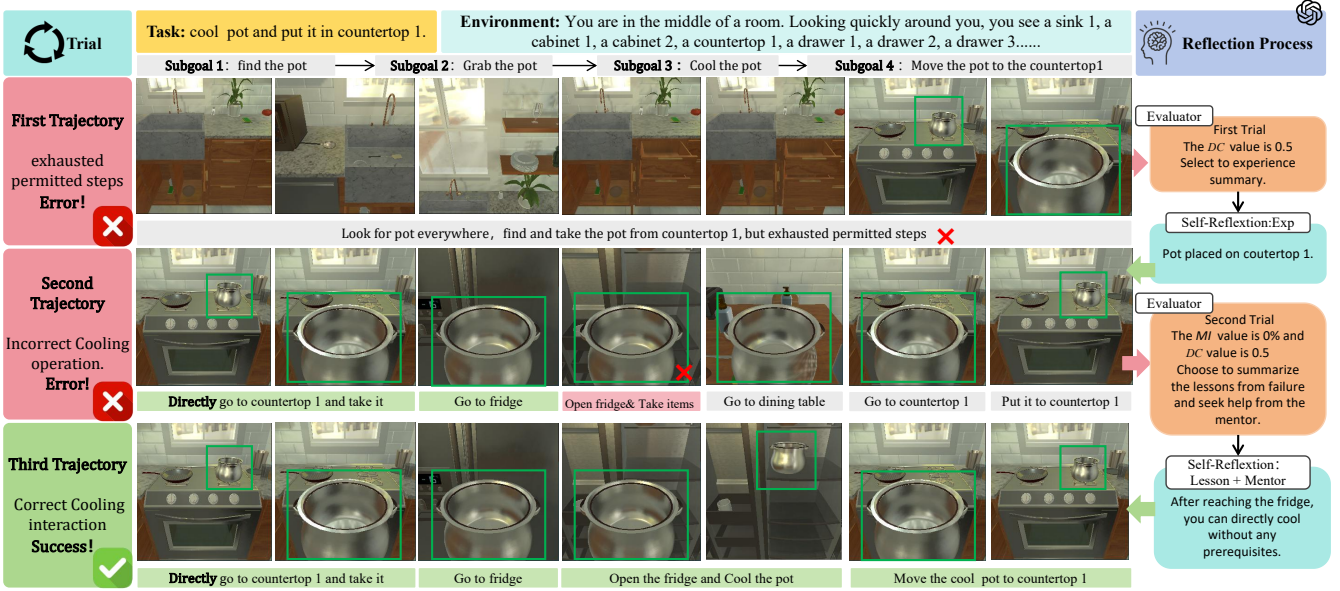


Fig. 4. In an ALFWorld example, FISRF corrects a failed trajectory by summarizing experiences in the first episode, subsequently learning from failures and seeking mentor guidance in the second episode.

where G represents the task goal, τ^t represents the trajectory of current trail t , Exp^t and $Lesson^t$ respectively represents the valuable experience and the precisely required failure lesson summarized through trail t , $M(q)$ represents the key knowledge from mentor, θ represents the parameters of SR_{sum} , mainly embodies through prompt.

E. The Flexible Reflection Process Under Mentor

A key innovation of our framework is the flexible, mentor-guided reflection process guided by difficulty of correction rather than difficulty of task itself. We assess this difficulty of correction based on task objectives, environmental context, and historical execution records, formulated as:

$$DC^t = \left(1 - \frac{Rate_{obj}^t + Rate_{inter}^t}{2}\right) \quad (7)$$

Where $Rate_{obj}^t$ represents the task completion rate of the current trial t , defined as $Rate_{obj}^t = \frac{Obj_{traj}^t}{Obj_{all}^t}$, where Obj_{all} represents the total number of sub-goals that the task needs to complete. $Rate_{inter}^t$ represents the current attempted item reach rate, defined as $Rate_{inter}^t = \frac{Inter_{traj}^t}{Inter_{all}^t}$, where $Inter_{all}$ represents the total number of items needed to be reached for the task. Concurrently, we define the magnitude of improvement based on difficulty of correction, quantifying the improvement between consecutive attempts on the same task as:

$$MI^t = \frac{DC^{t-1} - DC^t}{1 - DC^{t-1}} \times 100\% \quad (8)$$

where DC^{t-1} and DC^t denote the difficulty of correction in the current and previous rounds, respectively. As illustrated in Figure 3, the reflection strategy at trial t is determined by t , DC and MI^t : valuable experience summarization (Exp^t) triggers at Episode = 1 or $MI^t > 50\%$; failure lesson summarization ($Lesson^t$) triggers at Episode > 1 and $MI^t \leq 50\%$; When Episode = 1 with $DC > 0.5$

or Episode > 1 with $MI^t \leq 50\%$, the Agent is in the Zone of Proximal Development (ZPD), and mentor guidance $M(G, \tau^t)$ is triggered.

V. EXPERIMENTS

In this section, we evaluate FISRF in the household simulation environment AlfWorld [15] and further validate its effectiveness through real-world robotic experiments. Empirical results demonstrate that the proposed framework consistently outperforms baseline methods, achieving superior overall success rates as well as improvements across fine-grained performance metrics.

A. Experimental Setup

Environment and Dataset. We conduct our evaluation in **AlfWorld**, a text-based virtual household environment comprising five task categories: *Put*, *Clean*, *Heat*, *Cool*, and *Examine*. Our experiments span the complete set of 134 tasks across these categories. Each task is evaluated over five planning epochs to ensure statistical robustness and enhance the generalizability of the results.

Compared Methods. We benchmark FISRF against four representative categories of baselines in robotic planning and reflection: 1. **Plan-Only.** *ProgPrompt* [1], which directly generates executable action sequences with LLMs under the in-context learning paradigm [23], without explicit reasoning or reflection mechanisms. 2. **Reason-Only.** *ReAct* [6], which interleaves reasoning and action generation, enabling the LLM to iteratively decide subsequent actions through stepwise deliberation. 3. **Reason-Reflect.** *Reflexion* [5], which augments ReAct with a reflection module that leverages environmental feedback to analyze failures and refine future planning. 4. **Reason-Flexible Reflect.** *FCRF* [12], which further incorporates valuable historical experiences and adaptively adjusts reflection intensity according to task complexity, enabling more flexible and targeted self-reflection.

TABLE I

SUCCESS RATES (%) OF FISRF AND BASELINE METHODS ON ALFWORLD ACROSS DIFFERENT TASK CATEGORIES. RESULTS ARE REPORTED UNDER GEMINI 2.5 AND GPT-4o MODEL

Method	Success Rate (%)↑							
	Put	Clean	Heat	Cool	Examine	SR@1	SR@2	Overall
<i>Model: Gemini 2.5</i>								
Plan-Only	82.9	71.0	82.6	90.5	66.7	73.9	76.1	79.1
Reason-Only	90.2	80.6	79.2	95.7	55.6	76.1	82.1	83.6
Reason-Reflect	97.6	80.6	78.3	90.5	66.7	76.1	81.3	85.1
Reason-Flex Reflect	100.0	87.1	60.9	90.5	88.9	78.4	85.8	87.3
FISRF (Ours)	100.0	90.3	91.3	100.0	88.9	91.0	93.3	94.8
<i>Model: GPT-4o</i>								
Plan-Only	64.5	66.6	30.0	62.5	26.6	61.9	65.7	68.7
Reason-Only	86.3	87.5	77.1	93.3	58.3	80.6	81.3	82.8
Reason-Reflect	84.9	84.3	80.0	93.3	58.3	80.6	82.8	83.6
Reason-Flex Reflect	91.4	90.3	93.5	100.0	63.6	82.1	89.0	91.0
FISRF (Ours)	100.0	93.5	95.7	100.0	77.8	85.8	91.8	94.8

TABLE II

FLEXIBILITY AND EFFICIENCY OF REFLECTION METHODS. OUR FISRF DEMONSTRATES SIGNIFICANTLY HIGHER FLEXIBILITY AND EFFICIENCY.

Methods	Efficiency				Flexibility	
	Cost _{ref} @4 ↓	Rate _{rev} (%) @4 ↑	Round _{rev} @2 ↓	Round _{rev} @4 ↓	R ↑	CV(%) ↑
<i>Model : Gemini 2.5</i>						
Reason-Reflect	913.7	64.3	3.26	4.79	207	23.5
Reason-Flex Reflect	558.3	69.6	2.30	3.15	237	21.5
FISRF(Ours)	370.5	87.5	1.45	1.71	253	36.7
<i>Model : GPT 4o</i>						
Reason-Reflect	953.6	47.6	3.58	5.70	175	22.4
Reason-Flex Reflect	570.4	71.4	2.44	3.13	197	18.2
FISRF(Ours)	427.3	87.5	1.79	2.14	247	24.3

Metrics. We evaluate the methods from three aspects:

- **Success Rate** quantifies the overall effectiveness of the planning and reflection mechanisms, defined as the proportion of tasks successfully completed by the end of each trial.
- **Efficiency** is assessed through two indicators: **Reflection Cost** (Cost_{ref}): the average token cost per successful recovery, defined as $\text{Cost}_{\text{ref}} = \frac{\text{Cost}_{\text{all}}}{E_{\text{corrected}}}$, where Cost_{all} is the total token consumption across all episodes and E_{corrected} is the number of tasks recovered via reflection. **Revision Rate** (Rate_{rev}): the fraction of initially failed tasks that are successfully recovered, $\text{Rate}_{\text{rev}} = \frac{E_{\text{corrected}}}{E_{\text{initial}}}$, where E_{initial} denotes the number of failures in the initial planning phase. **Revision Round** (Round_{rev}): the mean number of reflection iterations required to successfully recover from a failed task, defined as $(\text{Round}_{\text{rev}}) = \frac{\text{Round}_{\text{all}}}{E_{\text{corrected}}}$, where Round_{all} the total number of reflection rounds accumulated across all episodes and E_{corrected} is the number of tasks recovered via reflection.
- **Flexibility** evaluates the adaptability of reflection length to task complexity, quantified by: **Range** (R): The

difference between the maximum number of tokens in a single reflection and the minimum number of tokens in a single reflection. **Coefficient of Variation** (CV): normalized dispersion, defined as $\text{CV} = \frac{\text{Cost}_{\text{ref,STD}}}{\text{Cost}_{\text{ref,AVE}}}$

B. Main Results

The main results are reported in Table I and Table II. To balance performance and computational cost, all experiments employ the official *GPT-4o* and *Gemini 2.5* models. Figure 4 presents a representative long-horizon task solved by FISRF. The key findings are summarized as follows: 1. FISRF achieves a 100% success rate on both the *Put* and *Cool* tasks. Compared with baseline methods, it yields performance gains of over 30% on the *Examine* task and more than 4% overall (Table I), demonstrating substantially enhanced reflection capability. These improvements can be attributed to systematic failure analysis, task-aware estimation of difficulty of correction, and the effective integration of knowledge distilled from historical trajectories together with critical guidance provided by mentors. 2. We observe no significant correlation between reflection token length and subsequent error-correction effectiveness. Moreover, re-

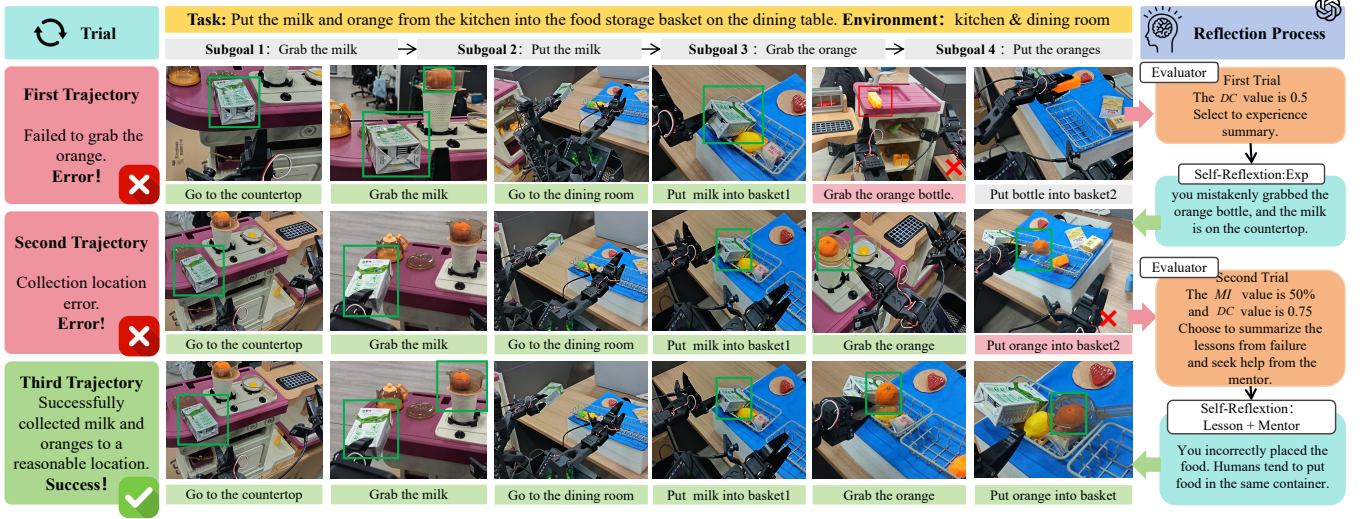


Fig. 5. Food collection task of a failed task successfully refined using our FISRF.

duced flexibility and efficiency in self-reflection consistently correspond to lower success rates. These observations support the necessity of adaptive reflection and validate the design of our proposed evaluation metrics. 3. Among all methods, the **Plan-Only** baseline yields the lowest success rate, highlighting the limitations of direct action generation without reasoning or reflection. **Reason-Only** and **Reason-Reflect** achieve stronger performance; however, although **Reason-Reflect** generally surpasses **Reason-Only**, it underperforms on certain tasks. This suggests that fixed-template reflection does not universally translate into performance gains. 4. In contrast to prior approaches, FISRF dynamically adjusts reflection intensity over a broader and more variable range (Table II). The metric R increases by over 6.8%, and the coefficient of variation (CV) rises by more than 8.5%, demonstrating enhanced **flexibility**. In terms of **efficiency**, FISRF achieves a higher revision rate and lower reflection cost, reflecting stronger error-correction capability and improved cost-effectiveness. Specifically, Rate_{ref} improves by more than 22.5% and Cost_{rev} decreases by over 25.1%, while $\text{Round}_{\text{rev}}$ decreases by over 31.6%. Overall, FISRF adaptively allocates computational resources, effectively leverages diverse execution feedback, and achieves superior performance–cost trade-offs.

VI. ANALYSIS AND DISCUSSION

A. Episode Analysis of Self-Reflection Process

We conduct an episode-level analysis to examine the underlying mechanisms and behavioral patterns of the self-reflection process (Figure 6). For each erroneous long-horizon task, the LLM is prompted to iteratively replan over five successive rounds, and task completion outcomes are recorded at each episode. The main observations are summarized as follows. 1. As illustrated in the left panel, FISRF consistently surpasses all baseline methods across episodes and exhibits faster convergence. This indicates superior robustness and performance, as well as improved computational efficiency in the iterative reflection process.

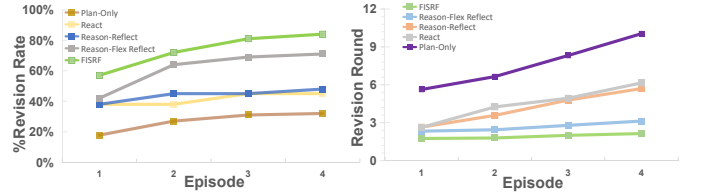


Fig. 6. Result of success rate and round revision with different episodes. Our FISRF outperforms in all episodes, converges more quickly and has a significant better efficiency.

2. Across all episodes, the **Plan-Only** method demonstrates the weakest error-recovery capability. While both **Reason-Only** and **Reason-Reflect** yield performance improvements, the gap between these two approaches remains marginal, suggesting limited additional benefit from fixed-pattern reflection. 3. With guidance from mentor-provided key knowledge, FISRF requires fewer reflection iterations to resolve errors, demonstrating significantly improved reflection efficiency and faster convergence. This further indicates that the judicious incorporation of mentor guidance, while preserving adaptive reflection flexibility, substantially enhances problem-solving capability and facilitates more effective self-reflective refinement.

B. Ablation Study

To comprehensively evaluate the contribution of each module and examine their interactions, we conduct ablation studies on the full ALFWORLD dataset. The detailed results are reported in Table III. In the **w/o Exp** variant, removing experience retention reduces the overall success rate by 7.9% and decreases R by 21.7%. This confirms that accumulated experience is critical for reflective adaptability and stable long-horizon planning. In the **w/o Fail** model, the variant that excludes lessons derived from failures results in a 6.3% decline in success rate, a 39.7% reduction in CV, and a 27.5% increase in reflection cost. These results underscore the importance of failure-informed knowledge in preserving reflective flexibility and ensuring cost-effective recovery in long-horizon planning.

TABLE III

ABLATION STUDY OF FISRF COMPONENTS IN ALFWORLD TASKS. ALL DESIGNED MODULES CONTRIBUTE TO THE PERFORMANCE.

Method	SR (%)	R (↑)	CV (↑)	Round _{rev} (↓)	Cost _{ref} (↓)	Rate _{rev} (↑)
Full	94.8	253	36.7	1.7	370.5	87.5
w/o Exp	87.3	198	24.5	3.6	487.3	68.7
w/o Lesson	88.8	208	22.1	2.4	472.5	75.3
w/o Mentor	82.9	272	27.6	4.2	550.7	63.5

In the **w/o Mentor** model, removing mentor guidance reduces the success rate by 12.6% and increases reflection cost by 48.6%, demonstrating its importance in improving performance and efficiency. Ablating any individual module further degrades flexibility and efficiency, leading to a lower overall success rate.

C. Real-World Robotic Experiment

We validate the proposed approach on a dual-arm mobile robot platform in real-world kitchen item manipulation tasks. As illustrated in Figure 5, the robot equipped with FISRF demonstrates enhanced self-reflection and recovery performance. Additional details will be provided on our website and in supplementary videos.

VII. CONCLUSIONS

Inspired by flexible instructional scaffolding theory in cognitive psychology, we propose **FISRF**, a mentor-agent reflection framework. FISRF determines whether the robot operates within its **Zone of Proximal Development(ZPD)** by jointly considering difficulty of correction and magnitude of improvement, and accordingly adapts reflection strategies or requests mentor guidance. It integrates experiential knowledge, failure feedback, and mentor-provided information, and incorporates a dedicated memory module for efficient reflection management in long-horizon LLM-based agents. Extensive experiments in virtual home and real-world environments demonstrate that FISRF improves planning flexibility, efficiency, and success rate, thereby enhancing the reliability of home robotic systems.

REFERENCES

- [1] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.
- [2] Jiatao Zhang, Lanling Tang, Yufan Song, Qiwei Meng, Haofu Qian, Jun Shao, Wei Song, Shiqiang Zhu, and Jason Gu. Fltrnn: Faithful long-horizon task planning for robotics with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6680–6686. IEEE, 2024.
- [3] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*, 2024.
- [4] Weiran Yao, Shelby Heinecke, Juan Carlos Nieves, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

- [5] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- [6] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.
- [7] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [8] Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-pro: Learning to evolve via policy-level reflection and optimization. *arXiv preprint arXiv:2402.17574*, 2024.
- [9] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.
- [10] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.
- [11] Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Generating instruction manuals by llm agents via interactive environmental learning. *arXiv preprint arXiv:2405.16247*, 2024.
- [12] Yufan Song, Jiatao Zhang, Zeng Gu, Qingmiao Liang, Tuocheng Hu, Wei Song, and Shiqiang Zhu. Fcrf: Flexible constructivism reflection for long-horizon robotic task planning with large language models. *arXiv preprint arXiv:2507.14975*, 2025.
- [13] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*, 2025.
- [14] Rachel R. Van Der Stuyf. Scaffolding as a teaching strategy. *Adolescent Learning and Development*, 52(3):5–18, 2002.
- [15] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworl: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.
- [16] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.
- [17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [18] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.
- [19] Haotian Sun, Yuchen Zhuang, Ling kai Kong, Bo Dai, and Chao Zhang. Adapllanner: Adaptive planning from feedback with language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [20] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- [21] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.
- [22] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Chatgpt empowered long-step robot control in various environments: A case application. *IEEE Access*, 2023.
- [23] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.