SkillGen: Learning Domain Skills for In-Context Sequential Decision Making

Ruomeng Ding[1*], **Wei Cheng**[2], **Minglai Shao**[3†], **Chen Zhao**[4]
[1]University of North Carolina at Chapel Hill
[2]NEC Laboratories America
[3]Tianjin University
[4]Baylor University
ruomeng@unc.edu, weicheng@nec-labs.com, shaoml@tju.edu.cn, chen_zhao@baylor.edu

Abstract

Large language models (LLMs) are increasingly applied to sequential decision-making through in-context learning (ICL), yet their effectiveness is highly sensitive to prompt quality. Effective prompts should meet three principles: focus on decision-critical information, provide step-level granularity, and minimize reliance on expert annotations through label efficiency. However, existing ICL methods often fail to satisfy all three criteria simultaneously. Motivated by these challenges, we introduce SkillGen, a skill-based ICL framework for structured sequential reasoning. It constructs an action-centric, domain-level graph from sampled trajectories, identifies high-utility actions via temporal-difference credit assignment, and retrieves step-wise skills to generate fine-grained, context-aware prompts. We further present a theoretical analysis showing that focusing on high-utility segments supports task identifiability and informs more effective ICL prompt design. Experiments on ALFWorld, BabyAI, and ScienceWorld, using both open-source and proprietary LLMs, show that SkillGen achieves consistent gains, improving progress rate by 5.9%–16.5% on average across models.

Code — https://github.com/ruomengd/SkillGen

1 Introduction

Large language models (LLMs) are increasingly applied to multi-step decision-making tasks across domains such as embodied control (Li et al. 2024; Yang et al. 2025), text-based games (Liu et al. 2024; Klissarov et al. 2024), and online shopping (Yao et al. 2022; Zhou et al. 2024b). These tasks require agents to operate in dynamic environments, interact with the world through sequences of actions, and pursue long-horizon goals. In contrast to supervised fine-tuning (SFT) methods (Chen et al. 2023; Zeng et al. 2024), which depend on large-scale expert demonstrations, in-context learning (ICL) offers a more lightweight and efficient alternative by guiding inference with only a few examples (Achiam et al. 2023). Consequently, ICL has emerged as a central reasoning paradigm in many LLM-based agent frameworks for decision-making (Yao et al. 2023; Shinn et al. 2023; Sun et al. 2023). To make ICL effective for multi-step tasks, prompt design should adhere to three key principles: (1) *Focused*: emphasize decision-critical information while minimizing redundant context; (2) *Granular*: offer fine-grained, step-level guidance that aligns to the evolving task state; (3) *Label-efficient*: replaces costly expert trajectories with subgoal completion and success signals, which offer a more scalable and structurally aligned form of supervision.

ICL Method	Focused	Granular	Label-efficient
Fixed Prompting	✗	✗	✓
Task-level Retrieval	✗	✗	✗
Step-wise Retrieval	✗	✓	✗
Insight Summary	✓	✗	✗
SkillGen (ours)	✓	✓	✓

Table 1: Comparison of ICL Methods.

Recent advances in ICL have enhanced performance by shifting from fixed, hand-crafted prompts to more context-sensitive prompt designs (Zhang, Feng, and Tan 2022; Liskavets et al. 2025). As summarized in Table 1, various methods target different aspects of prompt design. Fixed prompting relies on limited examples that ignore task-specific context. Although label-free, such prompts fail to provide actionable, decision-level guidance. Task-level retrieval methods, such as Synapse (Zheng et al. 2024), retrieve full expert demonstrations based on task metadata and use them as few-shot exemplars. While this improves contextual relevance, the retrieved prompts often include redundant steps and lack step-level resolution, limiting both focus and granularity. Step-level retrieval strategies, such as Trad (Zhou et al. 2024a), enhance granularity by retrieving trajectory fragments at each decision step. However, the retrieved actions are often disconnected and lack structural coherence, which undermines decision focus. These approaches often depend on expert-annotated trajectories, thereby reducing their label efficiency. Insight summarization approaches, including Leap (Zhang et al. 2024b) and ExpeL (Zhao et al. 2024), generate high-level insights by comparing correct and wrong solutions. Yet these summarized insights are often too abstract to support intermediate decisions, providing limited fine-grained guidance. While each of these approaches ad-

*The work was done while the author was affiliated with Georgia Institute of Technology
†Corresponding author
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dresses different aspects of prompt design, none of them simultaneously meets all three core principles.

To address these limitations, we propose SkillGen, an ICL framework that extracts and applies **domain-oriented** and **action-centric skills**. As illustrated in Figure 1, SkillGen operates in three stages, the first two stages are performed offline, while the third leverages the extracted skills to generate actions step-by-step during inference: (1) *Domain Knowledge Construction* – We construct an action-centric domain knowledge graph from sampled trajectories, effectively capturing the structural dynamics of the task. (2) *Domain Skill Extraction* – Temporal-Difference (TD) based credit assignment is employed to identify actions that consistently contribute to task progress; (3) *Skill-Based In-Context Learning* – During inference, SkillGen combines a golden segment with step-wise skills retrieved based on the current transition history to guide action generation. To support focused reasoning, SkillGen constructs prompts around decision-critical skill segments while filtering out irrelevant context. For fine-grained guidance, it encodes temporal and structural dependencies from the domain graph to retrieve skills that align with the current task history. To promote label efficiency, SkillGen leverages subgoal completion progress as weak supervision and applies TD-based action credits to extract skills, eliminating the need for full expert trajectories. SkillGen improves average progress rate by 5.9%–16.5% across ALFWorld, BabyAI, and ScienceWorld, showing clear gains in sequential decision-making. To conclude, our primary contributions are as follows:

- We address the challenge of designing in-context learning (ICL) prompts that jointly support decision focus, step-level granularity, and label efficiency in sequential decision-making tasks.
- We propose SkillGen, a framework that learns domain-oriented skills to support focused, fine-grained ICL. Theoretical analysis shows that high-utility segments enhance task identifiability and ICL prompts.
- Our empirical results show that SkillGen consistently improves progress and success rates across various tasks. Ablations show that both the golden segment and step-wise skill retrieval contribute to performance gains.

2 Related Work

Sequential Decision Making with LLMs. Recent advances in LLM-based decision-making have led to interactive agents that operate in multi-turn loops, either selecting actions directly or reasoning before acting (Yao et al. 2023; Zhao et al. 2025). To address long-horizon tasks, many methods incorporate feedback-driven refinement (Shinn et al. 2023; Sun et al. 2023; Chen et al. 2024a) or structured search (Besta et al. 2024; Zhuang et al. 2024; Li et al. 2025). Another line of work improves inference by retrieving expert or history information from offline interactions (Zheng et al. 2024; Zhou et al. 2024a). More recently, self-improving agents construct in-context examples from prior episodes, enabling generalization to unseen tasks without relying on expert demonstrations (Sarukkai, Xie, and Fatahalian 2025; Liu et al. 2025). These methods highlight a growing emphasis on experience-driven decision-making.

Knowledge-Augmented In-Context Learning. Knowledge augmented methods aim to enrich the prompt with structured information—such as relational or procedural knowledge, to provide stronger inductive bias and support more accurate reasoning. Some approaches guide reasoning with high-level prompts derived from prior interactions (Zhang et al. 2024a; Kong et al. 2025). Others inject retrieved graph-based knowledge to support multi-hop inference (Luo et al. 2024). LLMs can also self-synthesize reusable strategies from world models for generalization (Ding et al. 2024; Qiao et al. 2024b,a), or incorporate procedural knowledge via rule induction (Zhang et al. 2025) and skill reuse (Zhu et al. 2025; Chen et al. 2024b; Zhang et al. 2023; Zhao et al. 2024). These approaches enrich in-context learning by integrating external or derived task-specific knowledge as decision support.

3 Background

In-Context Learning. Xie et al. (Xie et al. 2022) model in-context learning (ICL) as an instance of implicit Bayesian inference. In this view, a language model infers a latent task parameter $\phi \in \Phi$ from the observed context $\mathcal{C}$ (e.g., a sequence of demonstrations or interaction history), forming a posterior $p(\phi \mid \mathcal{C})$ (Min et al. 2022; Falck, Wang, and Holmes 2024). Given a query input $x \in \mathcal{X}$, the model predicts by computing the posterior predictive distribution:

$$p(y \mid x, \mathcal{C}) = \int_{\phi} p(y \mid x, \phi)\, p(\phi \mid \mathcal{C})\, d\phi, \tag{1}$$

where $y \in \mathcal{Y}$ is the predicted output and $p(y \mid x, \phi)$ is the task-specific likelihood. Wies et al. (Wies, Levine, and Shashua 2023) formalize this intuition within a PAC framework by modeling pretraining as sampling from a latent task mixture $D = \sum_{\phi \in \Phi} \pi(\phi) P_\phi$, where $\pi(\phi)$ is the prior of latent task ϕ, and P_ϕ denotes the corresponding data distribution. In this view, ICL serves to recover the underlying task $\phi^\star$ from the prompt, enabling accurate prediction without updating model parameters.

LLMs for Sequential Decision-making. We consider adopting LLMs as autonomous agents for sequential decision-making. In such environments, agents cannot directly observe the underlying state. We model this setting as a Partially Observable Markov Decision Process (POMDP) (He et al. 2024; Sun et al. 2024), defined by POMDP $= (\mathcal{S}, \mathcal{A}, \Omega, P, R, O)$, where $\mathcal{S}$ denotes the latent state space, $\mathcal{A}$ the discrete action space, Ω the observation space, P the state transition function, R a sparse, progress-based reward function, and O the observation model. At each time step t, the agent receives a partial observation $o_t \sim O(o_t \mid s_t)$ and selects an action a_t based on the interaction transition: $h_t = \{(o_0, a_0), (o_1, a_1), \ldots, o_t\}$. Given the current history h_t, the LLM generates the next action through a prompting mechanism:

$$\pi(a_t \mid h_t) = \text{LLM}(a_t \mid \text{Prompt}(h_t)). \tag{2}$$

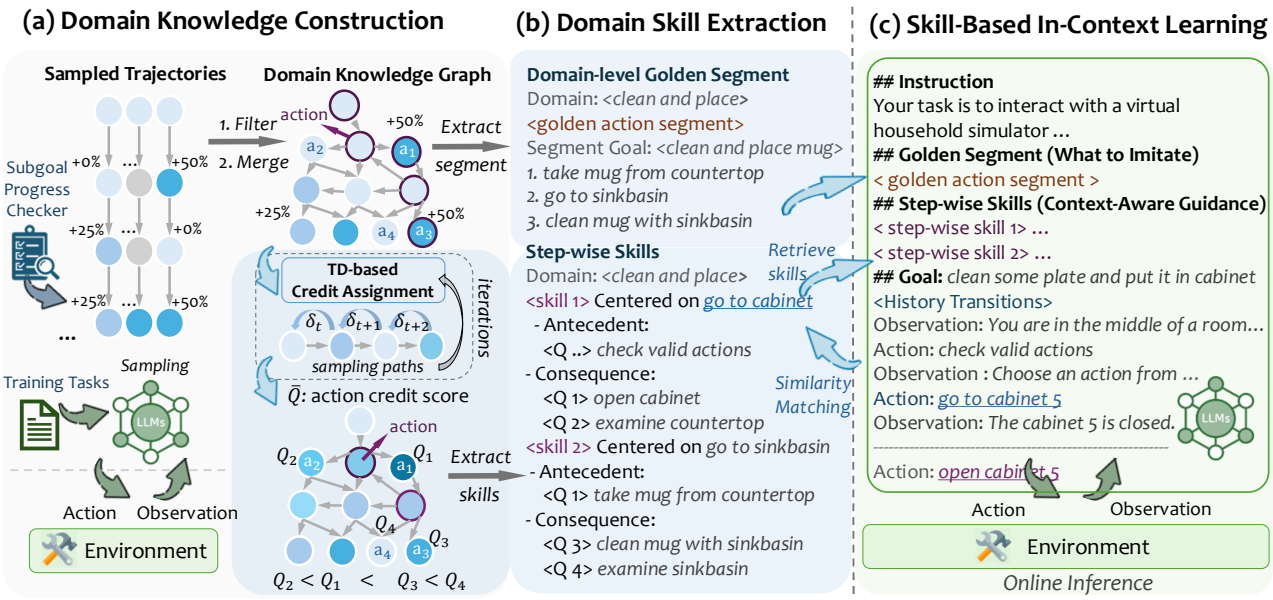


Figure 1: Framework of SkillGen.

This formulation enables LLMs to serve as decision-making agents in partially observable settings, leveraging contextual information without access to full environment states or explicit policy optimization. The agent aims to maximize cumulative task progress over long-horizon episodes.

4 Method

To enable focused and fine-grained in-context learning, we introduce two forms of decision-critical knowledge derived from structured action knowledge: (1) *Golden segment* – a concise action sequence extracted from training trajectories within the task domain, selected for its maximal contribution toward goal completion; (2) *Step-wise skills* – reusable local patterns centered on a key action, summarizing its typical antecedents and consequences within the domain.

4.1 Domain Knowledge Construction

To induce domain-level knowledge, we first sample diverse trajectories from LLMs using high-temperature stochastic decoding. Each training instance is denoted by $d = (m, g) \in \mathcal{D}_{\text{train}}$, where m is the task domain information and g is the task goal. For each instance, N trajectories are sampled:

$$\begin{aligned} \mathbb{T}_{\text{train}} &= \{(m, g, \mathcal{T}) \mid (m, g) \in \mathcal{D}_{\text{train}}\}, \\ \mathcal{T} &= \{(o_t, a_t, p_t)\}_{t=0}^{T}, \end{aligned} \tag{3}$$

where each step includes the observation o_t, action a_t, and progress signal p_t. To reduce noise and task-specific variance, trajectories are filtered by discarding invalid actions and those with zero final progress ($p_T = 0$). Actions are abstracted by removing object-specific identifiers (e.g., *"open cabinet 5"* $\rightarrow$ *"open cabinet"*) to reveal transferable patterns across instances, yielding sequences of action–progress pairs. The filtered trajectories for each domain m form:

$$\mathbb{T}_m = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}, \quad \text{with } \mathcal{T}_k = \{(a_t, p_t)\}_{t=0}^{T_k}. \tag{4}$$

From these, a directed domain knowledge graph $\mathbb{G}_m = (\mathbb{V}, \mathbb{E})$ is constructed, where each node $a \in \mathbb{V}$ denotes an action, and each edge $(a_i, a_j) \in \mathbb{E}$ indicates a transition between consecutive actions. All paths share a common start a_{start} and end node a_{end} to ensure structural consistency. Edges are annotated with sets of observed progress deltas:

$$\begin{aligned} \mathcal{P}_\Delta(a_i, a_j) = \{p_{t+1} - p_t \mid (a_t, p_t), (a_{t+1}, p_{t+1}) \in \mathcal{T}, \\ a_t = a_i,\ a_{t+1} = a_j\} \end{aligned} \tag{5}$$

To maintain graph quality, self-loops are removed and low-impact nodes pruned based on $\mathcal{P}_\Delta(a_i, a_j)$ once the graph exceeds a reasonable scale. The resulting graph captures reusable action patterns and domain-level decision dynamics, offering a structured foundation for skill extraction. We extract a *golden segment*—a short action sequence with the highest subgoal progress rate within the domain—to serve as a concise, decision-critical exemplar for focused prompting.

4.2 Domain Skill Extraction

The progress delta $\mathcal{P}_\Delta$, annotated on graph edges, provides sparse, subgoal-level feedback by marking major milestones such as completing a cleaning step or reaching a destination. However, it often fails to capture the contribution of intermediate actions that enable these outcomes. As illustrated in Figure 2, only a few steps in the task *"heat some apple and put it in countertop"* receive positive deltas (e.g., *heating and placing the apple*), while prerequisite actions—like navigating to and opening the microwave—remain uncredited despite being causally necessary. This illustrates the need for credit assignment to assign action value more accurately.

To address it, we treat $\mathcal{P}_\Delta$ as delayed rewards and propagate them backward along sampled action paths to estimate fine-grained action utility. While various metrics (Mesnard et al. 2021) provide alternative means of estimating action

importance, we adopt temporal difference learning with eligibility traces (TD(λ)) (Sutton 1988) to learn an action-value $Q(a)$, capturing the expected long-term credit of each action.

TD-based Credit Assignment. In the action-centric graph $\mathbb{G}_m = (\mathbb{V}, \mathbb{E})$, each iteration samples a set of action paths from a designated start node a_{start} to an end node a_{end} for downstream credit estimation. Formally, for iteration $i = 1, \ldots, N$, we sample:

$$\tau^{(i)} = (a_1^{(i)}, a_2^{(i)}, \ldots, a_{T_i}^{(i)}) \sim \mathcal{S}_{\text{path}}(a_{\text{start}}, a_{\text{end}}, \mathbb{G}_m), \quad (6)$$

where each $\tau^{(i)}$ is a valid path in $\mathbb{G}_m$ satisfying $(a_t^{(i)}, a_{t+1}^{(i)}) \in \mathbb{E}$, $a_1^{(i)} = a_{\text{start}}$, $a_{T_i}^{(i)} = a_{\text{end}}$. Here, $\mathcal{S}_{\text{path}}$ denotes a uniform distribution over all valid paths from a_{start} to a_{end} in $\mathbb{G}_m$. For each path $\tau = [a_0, a_1, \ldots, a_T]$, action credits are estimated using temporal-difference learning with eligibility traces. At each step t, the reward r_t is defined based on empirical progress deltas. If such records exist, i.e., $\mathcal{P}_\Delta(a_t, a_{t+1}) \neq \emptyset$, a value is uniformly sampled and perturbed with Gaussian noise:

$$r_t \sim \text{Uniform}(\mathcal{P}_\Delta(a_t, a_{t+1})) + \mathcal{N}(0, \sigma^2), \quad (7)$$

where $\text{Uniform}(\cdot)$ denotes a uniform distribution over finite set $\mathcal{P}_\Delta(a_t, a_{t+1})$. Otherwise, the reward defaults to pure noise, $r_t = \epsilon$, treating the transition as a step without observable progress. For each action a_t along path τ, the temporal-difference (TD) error is computed as:

$$\delta_t = r_t + \gamma Q(a_{t+1}) - Q(a_t). \quad (8)$$

To propagate credit backward, the eligibility trace for the current action a_t is incremented, and all actions with nonzero trace values are updated:

$$\begin{aligned} E(a_t) &\leftarrow E(a_t) + 1, \quad Q(a) \leftarrow Q(a) + \alpha \delta_t E(a), \\ E(a) &\leftarrow \gamma \lambda E(a), \quad \forall a \in \{a' \mid E(a') > 0\}. \end{aligned} \quad (9)$$

Here, α is the learning rate, γ the discount factor, and λ the trace decay rate. The eligibility trace $E(a)$ accumulates credit for recently visited actions, enabling delayed reward signals to influence earlier decisions across the sampled paths. After learning, $Q(a)$ values are normalized into a dense credit distribution:

$$\bar{Q}(a) = \frac{\max(Q(a), 0)}{\sum_{a'} \max(Q(a'), 0)}, \quad (10)$$

emphasizing both goal-reaching actions and the intermediate steps that enable them.

Skill Extraction. For each action node a in the domain graph, we extract a local skill centered on a using its immediate neighbors and credit scores. We define the *antecedent set* as the set of predecessor actions (incoming edges), and the *consequence set* as the set of successor actions (outgoing edges). Each action is assigned a normalized credit score from TD-based propagation, and both sets are ranked accordingly. As shown in Figure 1 (b), the resulting skill is formulated as:

$$\text{Skill}(a) = (a, \textit{Antecedents}(a), \textit{Consequences}(a)), \quad (11)$$

where a is the central action, and the ranked context captures its typical usage patterns. This structure supports context-aware retrieval of reusable action-centric skills.

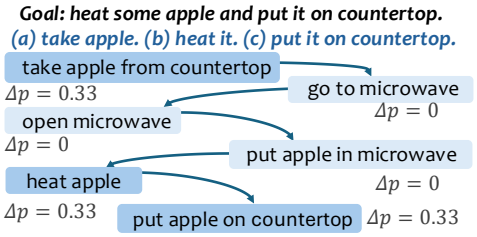


Figure 2: Sparse $\mathcal{P}_\Delta(a_i, a_j)$ reward only subgoal completions, omitting intermediate actions.

4.3 Skill-Based In-Context Learning

Building on structured skills extracted via TD-based credit assignment, we develop a retrieval-augmented prompting strategy that conditions a frozen LLM on both reusable domain knowledge and the agent's interaction history (Figure 1, right). The prompt integrates two levels of contextual information: (i) a domain-level *golden segment*—a concise, high-impact action sequence selected offline to serve as a focused exemplar; and (ii) *step-wise skills*—local, action-centric transitions retrieved based on the most recent action a_{t-1}.

Concretely, at each time step t, given a history transition $h_t = \{(o_0, a_0), \ldots, (o_{t-1}, a_{t-1}), o_t\}$, a pretrained semantic retriever processes the recent action a_{t-1} and retrieves the most relevant action $\hat{a}$ from the domain graph. The final prompt is formed by concatenating the golden segment from domain m, the retrieved skills $\text{Skill}(\hat{a})$, and the current trajectory history, each serialized into natural language. This composite prompt enables context-aware, credit-guided reasoning without requiring any parameter updates to the LLM. The next action is then sampled autoregressively:

$$\begin{aligned} \Phi &= \text{GoldenSegment}(m) \oplus \text{Skill}(\hat{a}), \\ \pi(a_t \mid h_t, \Phi) &= \text{LLM}(a_t \mid \text{Prompt}(h_t, \Phi)). \end{aligned} \quad (12)$$

4.4 Theoretical Analysis

In sequential reasoning tasks, prompts often include both decision-relevant and irrelevant segments, whereas uninformative tokens may obscure task-specific signals. Following the latent task mixture framework of Wies et al. (Wies, Levine, and Shashua 2023), we study how selecting a focused subset of high-utility content improves task identification. Assume prompts are drawn from a mixture distribution $\mathcal{D} = \sum_{\phi \in \Phi} \pi(\phi) P_\phi$, where ϕ indexes latent tasks, $\pi(\phi)$ is the prior, and P_ϕ the task-specific sequence distribution. Each input $x_t = (g, h_t)$ includes a goal g and history $h_t = \{(o_0, a_0), \ldots, o_t\}$, and $y_t = a_t$ is the action. We decompose the prompt as $p = p_{\text{focused}} \cup p_{\text{irrelevant}}$, defined relative to the ground-truth task $\phi^\star$: $p_{\text{focused}} := p_{\text{focused}} \mid \phi^\star$, capturing informative segments, and $p_{\text{irrelevant}} := p_{\text{irrelevant}} \mid \phi^\star$, capturing segments with little or misleading task evidence.

Theorem 1 (Task Identifiability). *Let $p \sim P_{\phi^\star}^{\otimes k}$ be a prompt sampled from the true task $\phi^\star \in \Phi$, and suppose it admits a decomposition $p = p_{focused} \cup p_{irrelevant}$, where the partition is defined relative to $\phi^\star$. Suppose that*

Method	Qwen2.5-7B-Instruct				Qwen-Turbo				GPT-4o-mini			
	GR	PR	SR	AUPC	GR	PR	SR	AUPC	GR	PR	SR	AUPC
0-shot	10.5	6.0	0.8	0.027	56.3	32.2	9.7	0.212	73.7	26.8	1.5	0.184
1-shot	28.1	16.0	2.2	0.095	63.9	55.3	36.5	0.380	77.3	43.3	10.5	0.292
Leap	27.7	21.2	5.2	0.125	66.3	55.6	37.3	0.386	78.6	50.8	11.2	0.348
Synapse (1-shot)	61.5	41.6	17.1	0.278	74.9	54.7	35.8	0.379	76.3	48.8	14.8	0.340
Synapse (3-shot)	71.4	44.8	19.4	0.302	78.4	60.6	47.0	0.421	77.4	52.9	17.8	0.360
Trad	65.4	44.2	22.4	0.296	65.5	54.8	35.8	0.372	79.1	49.1	16.4	0.341
SkillGen (ours)	**84.9**	**68.0**	**55.2**	**0.464**	**85.9**	**67.6**	**53.8**	**0.460**	**83.6**	**55.1**	**29.8**	**0.369**

Table 2: Grounding Rate [%] (↑), Progress Rate [%] (↑), Success Rate [%] (↑), and AUPC $[0, 1]$ (↑) on ALFWorld. The best method for each LLM is in **bold**; the second-best method is underlined.

Method	Qwen2.5-7B-Instruct				Qwen-Turbo				GPT-4o-mini			
	GR	PR	SR	AUPC	GR	PR	SR	AUPC	GR	PR	SR	AUPC
0-shot	31.8	21.8	7.1	0.037	50.2	32.7	19.6	0.092	55.3	34.2	22.3	0.129
1-shot	59.2	36.5	18.8	0.112	61.4	37.2	16.3	0.076	76.6	42.6	28.6	0.154
Leap	66.6	46.3	27.7	0.151	68.4	52.9	38.4	0.206	73.1	43.8	29.4	0.170
Synapse (1-shot)	67.2	39.4	21.4	0.153	65.0	55.8	45.5	0.242	86.5	44.9	33.9	0.169
Synapse (3-shot)	**78.6**	44.6	28.6	**0.163**	62.1	50.8	38.4	0.191	**92.8**	49.5	38.4	0.188
Trad	68.2	36.9	19.7	0.115	64.9	46.9	34.8	0.157	87.3	40.9	30.4	0.126
SkillGen (ours)	66.7	**50.0**	**31.2**	0.158	**73.9**	**59.4**	**45.5**	**0.254**	89.5	**57.6**	**41.1**	**0.248**

Table 3: Grounding Rate [%] (↑), Progress Rate [%] (↑), Success Rate [%] (↑), and AUPC $[0, 1]$ (↑) on BabyAI. The best method for each LLM is in **bold**; the second-best method is underlined.

Method	Qwen2.5-7B-Instruct				Qwen-Turbo				GPT-4o-mini			
	GR	PR	SR	AUPC	GR	PR	SR	AUPC	GR	PR	SR	AUPC
0-shot	10.8	27.1	9.0	0.136	**28.8**	19.3	4.4	0.113	34.3	44.3	7.7	0.206
1-shot	9.6	19.1	5.5	0.108	11.8	19.1	7.7	0.111	**34.3**	46.8	18.8	0.294
Leap	8.5	25.8	11.1	0.155	4.4	21.8	6.6	0.124	11.4	51.7	21.1	0.330
Synapse (1-shot)	8.6	15.4	4.4	0.090	5.4	16.0	3.3	0.106	14.0	52.8	25.6	0.334
Synapse (3-shot)	6.0	15.3	5.5	0.086	7.2	24.0	10.1	0.155	15.5	60.0	32.4	0.390
Trad	7.3	21.1	4.4	0.135	7.1	29.3	8.8	0.180	16.9	61.4	29.0	0.375
SkillGen (ours)	**16.1**	**46.7**	**23.4**	**0.298**	13.3	**37.7**	**11.1**	**0.242**	25.3	**67.3**	**40.2**	**0.442**

Table 4: Grounding Rate [%] (↑), Progress Rate [%] (↑), Success Rate [%] (↑), and AUPC $[0, 1]$ (↑) on ScienceWorld. The best method for each LLM is in **bold**; the second-best method is underlined.

$\min_{\phi \neq \phi^\star} \mathrm{KL}(P_{\phi^\star}(p) \| P_\phi(p)) > 8 \log\left(\frac{1}{c_1 \cdot c_2}\right)$. *Then, there exists a sample complexity threshold* $\tilde{m}_{\mathcal{D}} : (0,1)^2 \to \mathbb{N}$ *such that for any* $\epsilon, \delta > 0$*, if the number of in-context examples* $k \geq \tilde{m}_{\mathcal{D}}(\epsilon, \delta)$*, the following holds with probability at least* $1 - \delta$ *over the sampling of* p*:*

$$\forall \phi \neq \phi^\star, \quad \frac{P_\phi(p_{focused})}{P_{\phi^\star}(p_{focused})} \leq \frac{P_\phi(p)}{P_{\phi^\star}(p)} < \epsilon. \tag{13}$$

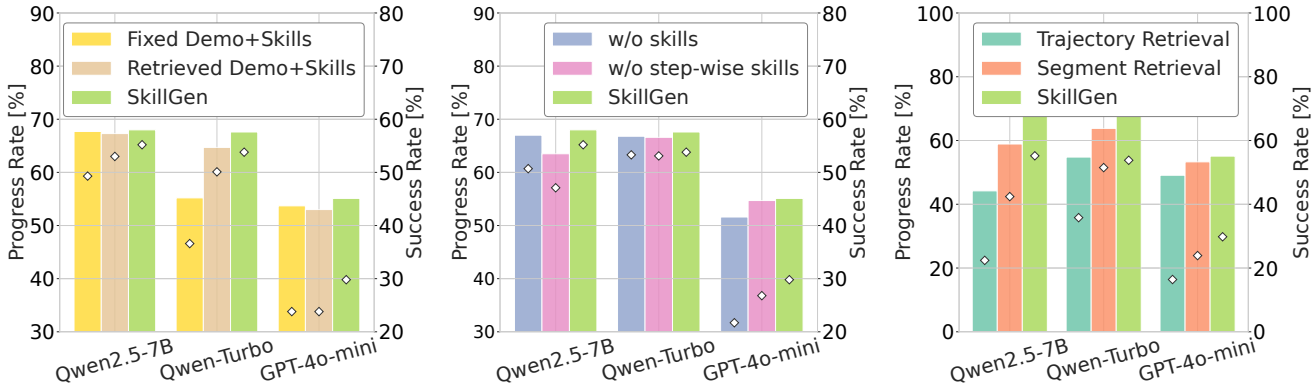
This result suggests that removing irrelevant segments sharpens the contrast between tasks, leading to a more task-aligned prompt with reduced ambiguity.

5 Experimental Setup

Datasets. We conduct experiments on three sequential decision-making datasets: ALFWorld (Shridhar et al. 2021), BabyAI (Chevalier-Boisvert et al. 2019), and ScienceWorld (Wang et al. 2022). These benchmarks span household tasks, grid-based navigation, and scientific reasoning, requiring agents to perform multi-turn interactions to achieve final goals. They cover diverse domains and increase in complexity—from ALFWorld to ScienceWorld. We employ four-fold cross-validation, and report results averaged over all folds. Based on the subgoal annotations provided by AgentBoard (Chang et al. 2024), we compute the subgoal achieved rate to quantify the model's step-wise progress. Notably, we use subgoal labels to construct domain graphs but do not rely on full expert trajectories during training or inference.

Evaluation Metrics. We evaluate all methods using four metrics: Grounding Rate (GR), measuring action validity in the current state; Progress Rate (PR), capturing the fraction of subgoals achieved; Success Rate (SR), indicating full task completion; and Area Under the Progress Curve (AUPC), which captures the cumulative task progress over time.

Baselines. We consider the following baselines: *0-shot* asks the agent to perform the task without any in-context examples. *1-shot* provides a single demonstration trajectory as example. *Leap* (Zhang et al. 2024b) enables the agent to self-revise by identifying and learning from mistakes in provided



(a) Golden Segment (b) Step-wise Skills (c) TD-based Credit Assignment

Figure 3: Ablation study of SkillGen on ALFWorld. Bars represent PR, ◊ markers indicate SR.

(a) Golden Segment (b) Step-wise Skills (c) TD-based Credit Assignment

Figure 4: Ablation study of SkillGen on ScienceWorld. Bars represent PR, ◊ markers indicate SR.

examples. *Synapse* (Zheng et al. 2024) retrieves and prompts entire expert trajectories from memory based on task metadata. *Trad* (Zhou et al. 2024a) guides inference by retrieving observation-action pairs from past interaction history. All baselines are built on the *Act* prompting framework (Yao et al. 2023), chosen for its simplicity and broad compatibility with instruction-following LLMs. To reduce foundation model bias, we evaluate all baselines using three models: Qwen2.5-7B-Instruct (Yang et al. 2024), Qwen-Turbo (Yang et al. 2024), and GPT-4o-mini (Hurst et al. 2024).

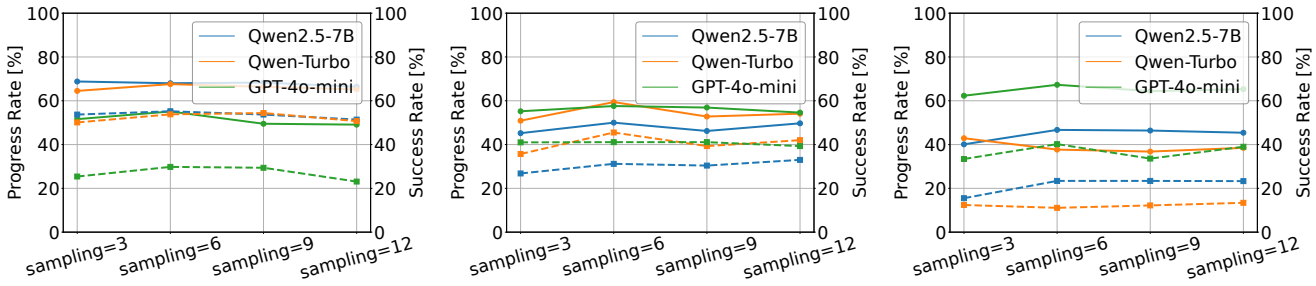
6 Evaluation Results

6.1 Main Results

ALFWorld. Table 2 presents the comparison of prompting strategies on ALFWorld. While SkillGen achieves the highest GR across all models, the most notable gains appear in PR and SR. On Qwen2.5-7B-Instruct, SkillGen achieves a PR of 68.0 and an SR of 55.2, substantially outperforming the strongest baseline, Synapse (3-shot), which achieves 44.8 (PR) and 19.4 (SR). On Qwen-Turbo, SkillGen reaches 67.6 (PR) and 53.8 (SR), surpassing the second-best Synapse (3-shot) with 60.6 and 47.0, respectively. Similar trends are observed on GPT-4o-mini, where SkillGen boosts SR from 17.8 to 29.8, again outperforming all alternatives. SkillGen outperforms Synapse and Trad by providing more reusable skills, yielding denser step-wise progress, higher subgoal completion, and the best AUPC for efficient task execution.

BabyAI. Table 3 shows that SkillGen achieves the highest PR scores across all models—reaching 59.4 on Qwen-Turbo and 57.6 on GPT-4o-mini. For SR, SkillGen shows significant improvements: a +7.0% gain on GPT-4o-mini (41.1 vs. 38.4). On Qwen2.5-7B-Instruct, SkillGen achieves 50.0 (PR) and 31.2 (SR), outperforming Synapse (3-shot) at 44.6 and 28.6, respectively. Compared to *Leap* and *Trad*, SkillGen achieves higher PR and SR—for example, +10.7 SR over *Trad* on GPT-4o-mini—highlighting the advantage of structured skill prompting. These results demonstrate that SkillGen outperforms both insight-summary and trajectory-retrieval baselines in spatially constrained environments.

ScienceWorld. In Table 4, SkillGen achieves a PR of 67.3 and an SR of 40.2 on GPT-4o-mini—a +7.8% improvement in SR over the strongest baseline, Synapse (3-shot), and



(a) ALFWorld (b) BabyAI (c) ScienceWorld

Figure 5: Sensitivity analysis of SkillGen with respect to the number of sampled trajectories per task for training set.

+11.2% over *Trad*—while also attaining the highest AUPC of 0.442. On Qwen-Turbo, the best baseline is *Trad* with 29.3 (PR) and 8.8 (SR), while SkillGen improves these to 37.7 and 11.1. Notably, naive baselines such as *0-shot* show inflated GR scores by repeatedly issuing generic queries like *"check valid action"*, which fail to make substantive progress. SkillGen consistently leads in AUPC, highlighting its efficiency in making steady progress. These results demonstrate that SkillGen 's structured prompting enables more goal-directed reasoning and higher task success.

6.2 Ablation Study

Effect of Golden Segment. To validate the effectiveness of *focused* prompting, we compare three strategies: (i) *Fixed Demo+Skills*, which uses a static, full demonstration augmented with skills; (ii) *Retrieved Demo+Skills*, which retrieves a relevant full trajectory from the training set. In Figure 3 (a) and 4 (a), *SkillGen* consistently outperforms the other two strategies across all tasks and model backbones. The largest gain is observed on ALFWorld with Qwen-Turbo, where SkillGen improves SR by +17.2% over Fixed and +3.7% over Retrieved. These results demonstrate that focused, high-impact segments lead to more effective decision-making by eliminating irrelevant context.

Effect of Step-wise Skills. To evaluate *granular* prompting, we compare *SkillGen w/o skills* and *SkillGen w/o step-wise retrieval*. As shown in Fig. 3(b) and 4(b), SkillGen performs best across all settings, improving SR over *w/o skills* by +8.1% (ALFWorld) and +4.5% (ScienceWorld). Step-wise retrieval provides further gains (+3.0% and +5.7%), while naive skill injection yields limited improvement, indicating that skill effectiveness depends on contextual relevance.

Effect of TD-based Credit Assignment. To assess the effect of TD-based credit assignment, we compare three step-wise retrieval methods: (i) *Trajectory Retrieval* (i.e., *Trad*), which retrieves step-wise observation–action pairs; (ii) *Segment Retrieval*, which retrieves action-only segments without applying credit assignment, using the original sparse progress signal. In Figure 3 (c) and 4 (c), SkillGen consistently achieves the highest performance, followed by *Segment Retrieval* with moderate gains. In ALFWorld, SkillGen achieves a SR of 55.2 (Qwen2.5-7B-Instruct), outperforming *Segment Retrieval* by +12.8 SR. On the more compositional ScienceWorld, SkillGen yields substantial gains, achieving a SR of 40.2 on GPT-4o-mini—a 16.8% increase over *Segment Retrieval*.

Dataset	Qwen2.5-7B		Qwen-Turbo		GPT-4o-mini	
	PR	SR	PR	SR	PR	SR
ALFWorld	63.2	50.1	64.6	52.2	32.9	14.2
BabyAI	43.2	32.1	50.5	36.6	56.8	41.1
ScienceWorld	29.1	18.9	27.7	10.0	56.2	32.4

Table 5: Results of SkillGen w/o subgoal annotations.

Effect of Subgoal Annotations. Table 5 shows that subgoal supervision mainly benefits complex tasks, improving ScienceWorld SR from 32.4% to 40.2% (GPT-4o-mini). Even without subgoal labels, SkillGen surpasses strong baselines, raising ALFWorld SR from 19.4% (*Synapse 3-shot*) to 50.1% with Qwen2.5-7B-Instruct, indicating robust interaction-driven skill learning.

Sampling Scale. We vary the number of sampled trajectories to test robustness. As shown in Fig. 5, performance is stable across settings. Even the smallest configuration (*sampling=3*) achieves strong SR (53.7% on ALFWorld with Qwen2.5-7B-Instruct). Increasing to *sampling=6* gives small gains, while larger values (*9,12*) add noise and can slightly reduce performance. This shows that a small, diverse trajectory set is sufficient for effective skill extraction.

7 Conclusion

In this study, we explore how to improve in-context learning for sequential decision-making by tackling three key challenges: maintaining decision focus, offering granular guidance, and reducing dependence on expert supervision. To address these issues, we propose SkillGen, a skill-aware prompting framework that leverages structured knowledge and weak supervision to enable fine-grained, context-sensitive reasoning. Our theoretical analysis highlights how decision-critical content supports task identifiability, motivating more principled prompt design. Experiments on ALFWorld, BabyAI, and ScienceWorld show consistent gains without expert demonstrations, highlighting SkillGen as a promising direction for improving generalization, efficiency, and coherence in LLM-based decision-making.

Acknowledgments

Ruomeng Ding, Wei Cheng, and Chen Zhao did not receive any financial support for this work, and Wei Cheng and Chen Zhao contributed only by developing the research ideas, participating in discussions, and providing feedback on the manuscript.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 16, 17682–17690.

Chang, M.; Zhang, J.; Zhu, Z.; Yang, C.; Yang, Y.; Jin, Y.; Lan, Z.; Kong, L.; and He, J. 2024. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. *Advances in Neural Information Processing Systems*, 37: 74325–74362.

Chen, B.; Shu, C.; Shareghi, E.; Collier, N.; Narasimhan, K.; and Yao, S. 2023. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Chen, L.; Tong, P.; Jin, Z.; Sun, Y.; Ye, J.; and Xiong, H. 2024a. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems*, 37: 37665–37691.

Chen, M.; Li, Y.; Yang, Y.; Yu, S.; Lin, B.; and He, X. 2024b. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37: 589–631.

Chevalier-Boisvert, M.; Bahdanau, D.; Lahlou, S.; Willems, L.; Saharia, C.; Nguyen, T. H.; and Bengio, Y. 2019. BabyAI: First steps towards grounded language learning with a human in the loop. In *International Conference on Learning Representations*, volume 105, 1–14. New Orleans, LA.

Ding, R.; Zhang, C.; Wang, L.; Xu, Y.; Ma, M.; Zhang, W.; Qin, S.; Rajmohan, S.; Lin, Q.; and Zhang, D. 2024. Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation. In *ACL (Findings)*.

Falck, F.; Wang, Z.; and Holmes, C. 2024. Is in-context learning in large language models bayesian? a martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, 12784–12805.

He, J.; Chen, S.; Zhang, F.; and Yang, Z. 2024. From Words to Actions: Unveiling the Theoretical Underpinnings of LLM-Driven Autonomous Systems. In *International Conference on Machine Learning*, 17807–17841. PMLR.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Klissarov, M.; Hjelm, D.; Toshev, A.; and Mazoure, B. 2024. On the Modeling Capabilities of Large Language Models for Sequential Decision Making. *arXiv preprint arXiv:2410.05656*.

Kong, M.; Wang, Z.; Shu, Y.; and Dai, Z. 2025. Meta-Prompt Optimization for LLM-Based Sequential Decision Making. *arXiv preprint arXiv:2502.00728*.

Li, D.; Zhao, X.; Yu, L.; Liu, Y.; Cheng, W.; Chen, Z.; Chen, Z.; Chen, F.; Zhao, C.; and Chen, H. 2025. SolverLLM: Leveraging Test-Time Scaling for Optimization Problem via LLM-Guided Search. *arXiv preprint arXiv:2510.16916*.

Li, M.; Zhao, S.; Wang, Q.; Wang, K.; Zhou, Y.; Srivastava, S.; Gokmen, C.; Lee, T.; Li, E. L.; Zhang, R.; et al. 2024. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37: 100428–100534.

Liskavets, B.; Ushakov, M.; Roy, S.; Klibanov, M.; Etemad, A.; and Luke, S. K. 2025. Prompt compression with context-aware sentence encoding for fast and improved llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 24595–24604.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. 2024. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*.

Liu, Y.; Si, C.; Narasimhan, K.; and Yao, S. 2025. Contextual Experience Replay for Self-Improvement of Language Agents. *arXiv preprint arXiv:2506.06698*.

Luo, L.; Zhao, Z.; Gong, C.; Haffari, G.; and Pan, S. 2024. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*.

Mesnard, T.; Weber, T.; Viola, F.; Thakoor, S.; Saade, A.; Harutyunyan, A.; Dabney, W.; Stepleton, T. S.; Heess, N.; Guez, A.; et al. 2021. Counterfactual Credit Assignment in Model-Free Reinforcement Learning. In *International Conference on Machine Learning*, 7654–7664. PMLR.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11048–11064.

Qiao, S.; Fang, R.; Zhang, N.; Zhu, Y.; Chen, X.; Deng, S.; Jiang, Y.; Xie, P.; Huang, F.; and Chen, H. 2024a. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37: 114843–114871.

Qiao, S.; Zhang, N.; Fang, R.; Luo, Y.; Zhou, W.; Jiang, Y.; Lv, C.; and Chen, H. 2024b. AutoAct: Automatic Agent Learning from Scratch for QA via Self-Planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 3003–3021.

Sarukkai, V.; Xie, Z.; and Fatahalian, K. 2025. Self-Generated In-Context Examples Improve LLM Agents for Sequential Decision-Making Tasks. *arXiv preprint arXiv:2505.00234*.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.

Shridhar, M.; Yuan, X.; Cote, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 1–14.

Sun, H.; Zhuang, Y.; Kong, L.; Dai, B.; and Zhang, C. 2023. Adaplanner: Adaptive planning from feedback with language models. *Advances in neural information processing systems*, 36: 58202–58245.

Sun, L.; Jha, D. K.; Hori, C.; Jain, S.; Corcodel, R.; Zhu, X.; Tomizuka, M.; and Romeres, D. 2024. Interactive planning using large language models for partially observable robotic tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 14054–14061. IEEE.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44.

Wang, R.; Jansen, P.; Côté, M.-A.; and Ammanabrolu, P. 2022. ScienceWorld: Is your Agent Smarter than a 5th Grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11279–11298. Association for Computational Linguistics.

Wies, N.; Levine, Y.; and Shashua, A. 2023. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36: 36637–36651.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yang, R.; Chen, H.; Zhang, J.; Zhao, M.; Qian, C.; Wang, K.; Wang, Q.; Koripella, T. V.; Movahedi, M.; Li, M.; et al. 2025. EmbodiedBench: Comprehensive Benchmarking Multi-modal Large Language Models for Vision-Driven Embodied Agents. *arXiv preprint arXiv:2502.09560*.

Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35: 20744–20757.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*, 1–14.

Zeng, A.; Liu, M.; Lu, R.; Wang, B.; Liu, X.; Dong, Y.; and Tang, J. 2024. AgentTuning: Enabling Generalized Agent Abilities for LLMs. In *Findings of the Association for Computational Linguistics*, 3053–3077.

Zhang, J.; Zhang, J.; Pertsch, K.; Liu, Z.; Ren, X.; Chang, M.; Sun, S.-H.; and Lim, J. J. 2023. Bootstrap Your Own Skills: Learning to Solve New Tasks with Large Language Model Guidance. In *Conference on Robot Learning*, 302–325. PMLR.

Zhang, T.; Madaan, A.; Gao, L.; Zhang, S.; Mishra, S.; Yang, Y.; Tandon, N.; and Alon, U. 2024a. In-Context Principle Learning from Mistakes. In *ICML 2024 Workshop on In-Context Learning*.

Zhang, T.; Madaan, A.; Gao, L.; Zheng, S.; Mishra, S.; Yang, Y.; Tandon, N.; and Alon, U. 2024b. In-context principle learning from mistakes. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zhang, Y.; Feng, S.; and Tan, C. 2022. Active Example Selection for In-Context Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9134–9148.

Zhang, Y.; Xiao, P.; Wang, L.; Zhang, C.; Fang, M.; Du, Y.; Puzyrev, Y.; Yao, R.; Qin, S.; Lin, Q.; Pechenizkiy, M.; Zhang, D.; Rajmohan, S.; and Zhang, Q. 2025. RuAG: Learned-rule-augmented Generation for Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhao, Q.; Li, D.; Liu, Y.; Cheng, W.; Sun, Y.; Oishi, M.; Osaki, T.; Matsuda, K.; Yao, H.; Zhao, C.; Chen, H.; and Zhao, X. 2025. Uncertainty Propagation on LLM Agent. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6064–6073. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Zheng, L.; Wang, R.; Wang, X.; and An, B. 2024. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. In *The Twelfth International Conference on Learning Representations*.

Zhou, R.; Yang, Y.; Wen, M.; Wen, Y.; Wang, W.; Xi, C.; Xu, G.; Yu, Y.; and Zhang, W. 2024a. TRAD: Enhancing LLM Agents with Step-Wise Thought Retrieval and Aligned Decision. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3–13.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2024b. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*, 1–14.

Zhu, Y.; Qiao, S.; Ou, Y.; Deng, S.; Lyu, S.; Shen, Y.; Liang, L.; Gu, J.; Chen, H.; and Zhang, N. 2025. KnowAgent: Knowledge-Augmented Planning for LLM-Based Agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 3709–3732. Association for Computational Linguistics. ISBN 979-8-89176-195-7.

Zhuang, Y.; Chen, X.; Yu, T.; Mitra, S.; Bursztyn, V.; Rossi, R. A.; Sarkhel, S.; and Zhang, C. 2024. ToolChain*: Efficient Action Space Navigation in Large Language Models with A* Search. In *The Twelfth International Conference on Learning Representations*, 1–14.