

Ability-Aligned Skill Evolution for LLM Agents via Reinforcement Learning

Fei Huang^{1,*}, Tianyu Chen^{1,*}, Zhongyu He¹, Yuanfan Li¹, Meng Hsuan Yu^{1,†},
Xingyang Li^{1,†}, Lu Pan¹, Ke Zeng¹, Xunliang Cai¹,

¹Meituan

Email: huangfei37@meituan.com, chentianyu18@meituan.com, lixingyang04@meituan.com

Abstract

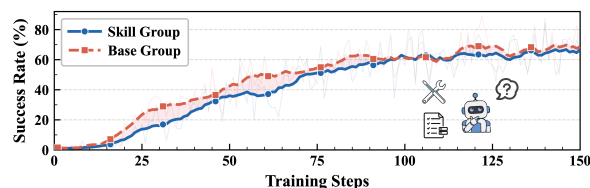
Large Language Model agents augmented with external skill banks have demonstrated remarkable performance on sequential decision-making tasks. However, existing methods update skills in an undirected, unverified manner and prematurely penalize agents for failing to utilize immature skills. We introduce a structured cognitive representation that decomposes both general and task-specific skills into four dimensions—D-PLAN, D-TRACK, D-EXEC, and D-GUARD—enabling precise failure attribution. Building upon this representation, we propose A^2SE , an ability-aligned skill evolution framework that synchronizes skill bank updates with the agent’s learning trajectory. Specifically, A^2SE concentrates evolution on the identified bottleneck dimension, safeguards updates via a causal verification mechanism with per-task-type rollback, and employs a competence-aligned skill reward to alleviate premature penalization. Extensive experiments on ALFWorld and WebShop demonstrate that A^2SE achieves 11.0% and 7.1% improvements over the strongest baselines, respectively, with ablation studies confirming the necessity of each proposed component.

1 Introduction

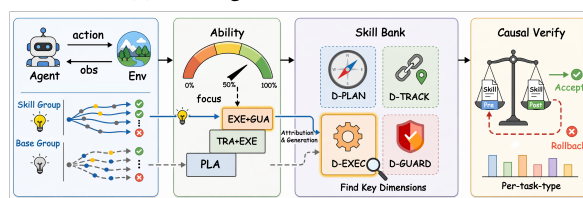
Agentic reinforcement learning has emerged as a powerful paradigm for training large language models as autonomous decision-makers, adeptly handling tasks ranging from interactive embodied environments (Shridhar et al., 2020; Feng et al., 2026) to complex web navigation (Yao et al., 2022). To mitigate the inefficiencies of learning from scratch under long-horizon partial observability, recent advancements increasingly augment agentic reinforcement learning with external *skill banks*—structured memory repositories that distill past experiences into reusable behavioral priors. As

*These authors contributed equally to this work.

†Corresponding author.



(a) Training Curves of Success Rate



(b) A^2SE Framework

Figure 1: (a) Skill group vs. Base group success rates under GRPO on WebShop, revealing the optimization misalignment in early training. (b) The A^2SE framework decomposes skills into four cognitive dimensions and dynamically focuses updates on the bottleneck dimension aligned with agent ability, safeguarded by causal verification with per-task-type rollback.

demonstrated by frameworks such as ARISE (Li et al., 2026b) and ASDA (Yim et al., 2026), co-evolving a dynamic skill bank alongside policy optimization empowers agents to bypass repetitive exploration and improve task success rates.

Despite these promising advances, we identify two fundamental misalignments that undermine existing skill-augmented RL paradigms. The first is a **competence misalignment** between injected knowledge and agent capability. Prior efforts have explored various skill categorizations to improve organization: SkillRL (Xia et al., 2026) distinguishes general from task-specific skills, D2Skill (Tu et al., 2026) separates task-level from step-level retrieval, and SkillX (Wang et al., 2026a) stratifies skills into strategic, functional, and atomic layers. However, these taxonomies remain *coarse-grained functional partitions*—they categorize *when* or *at what granularity* to invoke skills, but fail to decompose *what*

cognitive capability a skill provides. Consequently, each skill is still treated as an unstructured monolithic text block, precluding precise failure attribution when the agent errors. This limitation becomes particularly consequential because, as our observations reveal, agent failure modes shift systematically during training: from high-level planning errors in early stages to fine-grained constraint violations later on. Without the ability to attribute failures to specific cognitive dimensions, injected knowledge remains fundamentally misaligned with the agent’s evolving competence needs (Bengio et al., 2009).

The second is an **optimization misalignment** during policy updates. As the agent has not yet learned to utilize immature skills, the skill group underperforms the base group during early training (Figure 1(a)). Under GRPO’s group-relative advantage mechanism, this gap tends to assign lower advantages to skill-injected trajectories, discouraging the policy from following skill prompts. Consequently, the model struggles to assimilate the knowledge introduced by skill augmentation precisely when it has yet to internalize the skill style and injected knowledge.

Motivated by these insights, we propose **A²SE** (Figure 1(b)), an ability-aligned skill evolution framework that resolves both misalignments by synchronizing skill bank evolution with the agent’s learning trajectory. To address the competence misalignment, A²SE introduces a *structured cognitive representation* that decomposes both general and task-specific skills into four dimensions—D-PLAN, D-TRACK, D-EXEC, and D-GUARD—enabling precise failure attribution. Building on this representation, a *stage-aware dimension focus* mechanism targets skill updates at the agent’s current capability bottleneck, safeguarded by a *causal verification mechanism with per-task-type rollback* that ensures only empirically validated updates are retained, even when LLM-generated content hallucinates. To break the optimization misalignment, a *competence-aligned skill reward* dynamically shapes the reward distribution—asymmetrically amplifying sporadic successes and softening failure penalties during early exploration, then naturally decaying as the agent masters skill utilization, thereby restoring standard GRPO optimization. We experimentally validate A²SE on ALFWorld and WebShop, achieving 88.3% and 79.7% success rates respectively, surpassing the strongest skill-augmented baselines. Ablations confirm that

stage-aware dimension focus, competence-aligned reward, and causal verification are all essential for effective skill augmentation.

The main contributions of this work are summarized as follows:

- **Ability-diagnostic skill representation.** We introduce a cognitive skill decomposition that represents reusable agent skills along four dimensions: D-PLAN, D-TRACK, D-EXEC, and D-GUARD. Unlike coarse task-level or step-level skill taxonomies, this representation identifies which cognitive capability a skill supports and enables dimension-level failure attribution as the agent’s bottlenecks shift during training.
- **Ability-aligned skill evolution framework:** We propose A²SE, which dynamically synchronizes skill bank updates with agent ability by scheduling dimension-focused updates, employing causal verification with per-task-type rollback, and shaping rewards to facilitate skill acquisition during early training.
- **Comprehensive experimental validation:** Extensive experiments on ALFWorld and WebShop demonstrate that A²SE achieves 11.0% and 7.1% improvements over the strongest skill-augmented baselines, with ablation studies confirming the necessity of each component.

2 Related Work

2.1 LLM Agents for Sequential Decision-Making

The emergence of capable LLMs has catalyzed rapid development in autonomous agent systems for sequential decision-making (Wei et al., 2026). Early paradigms such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023) establish the foundation of interleaving reasoning with environment actions and learning from self-reflection, yet remain fundamentally context-bound—treating each episode in isolation with no persistent knowledge across task boundaries. To overcome this limitation, a growing line of work equips agents with external memory that supports continual adaptation beyond parameter updates (Zhao et al., 2024; ang Gao et al., 2026), exploring perspectives ranging from retention and forgetting (Chhikara et al., 2025) to structured organization and retrieval-aware optimization (Xu et al., 2026; Zhou et al.,

2025). Beyond storing raw interaction history, recent studies increasingly abstract experience into reusable knowledge such as reasoning strategies (Zhao et al., 2024) and hierarchical experience libraries (Yang et al., 2025b). Among these abstractions, skills have emerged as an especially effective form for improving policy performance on complex agentic tasks (Wang et al., 2026c), as they compress experience into concise, actionable guidance rather than verbose trajectory storage.

2.2 Skill-Augmented Agentic Reinforcement Learning

Skills serve as a non-parametric complement to RL by providing reusable guidance that can be retrieved into the policy context to improve task performance (ang Gao et al., 2026). Recent methods explore diverse strategies for building and maintaining skill banks: SkillRL (Xia et al., 2026) co-evolves a hierarchical skill bank via recursive failure synthesis from validated trajectories; D2Skill (Tu et al., 2026) generates task-level and step-level skills from training-time experience with utility-aware retrieval and pruning; EvolveR (Wu et al., 2026) and MemRL (Zhang et al., 2026) demonstrate the value of dynamically updating knowledge alongside policy optimization; SKILL0 (Lu et al., 2026) introduces a training-time curriculum that progressively withdraws skill context. Complementing these advances, Skills-Bench (Li et al., 2026a) provides a benchmark for evaluating agent skills across diverse tasks with curated skills and deterministic verifiers.

3 Preliminaries

3.1 Problem Setup

We formulate the sequential decision-making task as a Markov decision process (MDP) (Knox and Stone, 2012) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $s_t \in \mathcal{S}$ is the textual observation state, $a_t \in \mathcal{A}$ is the action space, $(s_{t+1}, r_t) \sim \mathcal{P}(\cdot | s_t, a_t)$ is the environment transition function, and $r_t \in \{0, 1\}$ is a sparse binary reward indicating task completion. An LLM-based agent π_θ interacts with the environment for up to T steps per episode.

The skill bank $\mathcal{B} = \mathcal{B}_{\text{gen}} \cup \mathcal{B}_{\text{spec}}$ comprises general skills shared across task types and task-specific skills. Each skill $s \in \mathcal{B}$ is retrieved via dense embedding matching: given a task description q , the top- k relevant skills are selected by $s^* = \arg \max_{s \in \mathcal{B}} \cos(E(q), E(\text{title}(s)))$, where E is

a sentence embedding model. At each training step, we sample a batch of M tasks; each task m is repeated N times, forming a task group $G_m = \{\tau_{m,1}, \dots, \tau_{m,N}\}$. The N rollouts are partitioned via a binary mask $\mathbf{m} \in \{0, 1\}^N$ into the *skill group* $\mathcal{G}_{\text{skill}}^m = \{j : m_j = 1\}$ and the *base group* $\mathcal{G}_{\text{base}}^m = \{j : m_j = 0\}$.

For task group G_m , we define group-level success rates as:

$$\text{SR}_{\text{skill}}^m = \frac{1}{|\mathcal{G}_{\text{skill}}^m|} \sum_{j \in \mathcal{G}_{\text{skill}}^m} R_j \quad (1)$$

$$\text{SR}_{\text{base}}^m = \frac{1}{|\mathcal{G}_{\text{base}}^m|} \sum_{j \in \mathcal{G}_{\text{base}}^m} R_j \quad (2)$$

where $R_j \in \{0, 1\}$ is the episodic completion reward.

The batch-level and per-task-type- c counterparts are denoted similarly. We further define the *skill advantage* $\Delta_m = \text{SR}_{\text{skill}}^m - \text{SR}_{\text{base}}^m$ and the *group reward baseline* $\mu_m = \frac{1}{N} \sum_{j=1}^N R_j$, which are shared across CASR and GRPO in subsequent sections.

3.2 Group Relative Policy Optimization

The policy is optimized via Group Relative Policy Optimization (Shao et al., 2024). For each task group G_m with rewards $\{R_j\}_{j=1}^N$, the advantage is normalized as:

$$\hat{A}_j = \frac{R_j - \mu_m}{\sigma_m} \quad (3)$$

where μ_m and σ_m are the mean and standard deviation of rewards within G_m . The GRPO objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E} \left[\frac{1}{N} \sum_{j=1}^N \min \left(\rho_j \hat{A}_j, \text{clip}(\rho_j, 1 - \epsilon, 1 + \epsilon) \hat{A}_j \right) - \delta \cdot D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \quad (4)$$

where $\rho_j = \frac{\pi_\theta(\tau_j)}{\pi_{\text{ref}}(\tau_j)}$ is the importance ratio, ϵ is the clipping coefficient, and δ controls the KL penalty strength. In §4.2.3, we extend GRPO by introducing a shaped reward $\tilde{R}_j$ that modulates advantage computation to facilitate skill acquisition in early training.

4 Methodology

As illustrated in Figure 2, we present **A²SE**, an ability-aligned skill evolution framework that dynamically synchronizes skill bank updates with

agent ability. The framework operates in two stages: first, it represents skills using a four-dimensional cognitive decomposition (§4.1); second, it aligns skill evolution with agent ability through dimension-focused updates, causal verification, and reward shaping (§4.2). The complete training pipeline is summarized in §4.3.

4.1 Four-Dimensional Cognitive Skill Representation

Existing skill-augmented frameworks treat each skill as an unstructured monolithic text block, leading to inefficient context usage and imprecise failure attribution. We design a structured skill representation by drawing upon cognitive science and empirical observations of LLM agent failures. Our design is grounded in two key insights: first, cognitive taxonomies from educational psychology (Krathwohl, 2002) decompose human knowledge into four types (conditional, declarative, procedural, and metacognitive) that humans acquire through distinct pathways; second, empirical analyses of LLM agent failures reveal systematic error patterns that shift during training, from high-level planning errors in early stages to fine-grained execution and constraint violations later (Shinn et al., 2023; Zhu et al., 2025).

Converging these perspectives, we decompose both general and task-specific skills into four meta-dimensions that align with these cognitive types and the dominant failure modes observed across sequential decision-making tasks: **D-PLAN (Planning & Exploration)** provides conditional knowledge for goal decomposition and search heuristics that guide the agent to establish objective hierarchies and actively explore uncovered paths, critical in early training when agents lack basic exploration strategies (Wang et al., 2023); **D-TRACK (State Tracking)** captures declarative knowledge for continuously tracking object states, locations, and attributes to prevent state loss in long-horizon tasks, addressing a distinct class of failures that persist independently from planning and execution errors (Yu et al., 2026; Tu et al., 2026); **D-EXEC (Execution Details)** supplies procedural knowledge specifying the precise format and fine-grained operation norms for specific actions, cognitively distinct from declarative knowledge and showing maximal effectiveness in later training stages (Wang et al., 2026b); and **D-GUARD (Error Guarding)** provides metacognitive knowledge defining negative boundaries and explicit constraints against com-

mon failure modes, prescribing what agents should avoid rather than what they should do, with prior work demonstrating independent contributions to overall performance (Xia et al., 2026).

Formally, each skill $s \in \mathcal{B}$ is structured as a JSON object comprising `task_type`, `skill_id`, `title`, `when_to_apply`, and a `meta_dimensions` dictionary that maps the four dimensions to their respective textual content (or null if inactive). The non-null entries, collectively denoted as `active_dimensions`, are synthesized by an LLM into a unified `skill_description` for prompt injection. This structured representation enables precise failure attribution, permits dimension-selective updates aligned with the agent’s current learning stage, and supports automatic pruning of dimensions that have been internalized by the policy.

4.2 A²SE: Ability-Aligned Skill Evolution

We synchronize skill bank updates with agent ability by scheduling dimension-focused updates aligned with the agent’s learning stage, employing causal verification to prevent skill pollution, and shaping rewards to facilitate skill acquisition during early training.

4.2.1 Stage-Aware Dimension Focus

We define the agent ability $A(t)$ as an EMA-smoothed, self-weighted combination of both groups:

$$A(t) = \text{EMA}_\beta[\gamma(t) \cdot \text{SR}_{\text{base}}(t) + (1 - \gamma(t)) \cdot \text{SR}_{\text{skill}}(t)] \quad (5)$$

The adaptive weight $\gamma(t) = \text{SR}_{\text{base}}(t)$ progressively shifts emphasis from skill-augmented performance in early training to intrinsic capability in later stages. The focus dimension set $\mathcal{F}(t)$ maps the agent’s developmental stage to the most impactful cognitive dimension:

$$\mathcal{F}(t) = \begin{cases} \{\text{PLA}\} & A(t) < \theta_1 \\ \{\text{TRA}, \text{EXE}\} & \theta_1 \leq A(t) < \theta_2 \\ \{\text{EXE}, \text{GUA}\} & A(t) \geq \theta_2 \end{cases} \quad (6)$$

During the LLM-driven skill update phase, a two-stage prompting strategy is employed: Phase 1 attributes the root cause of failed trajectories to a specific dimension; Phase 2 generates updated content strictly constrained to dimensions in $\mathcal{F}(t)$. A dual-filtering mechanism of prompt constraint and parse rejection ensures that any update targeting $d \notin \mathcal{F}(t)$ is discarded, preventing premature or redundant knowledge injection.

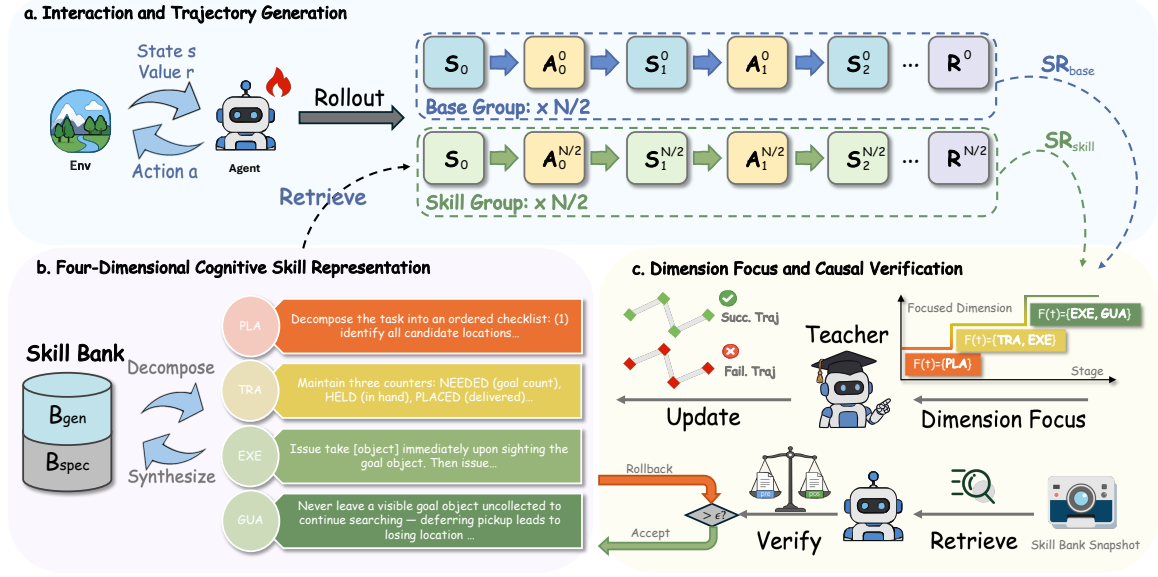


Figure 2: **Overview of the A²SE framework.** The framework comprises three components: (a) interaction and trajectory generation where the agent interacts with the environment and rollouts are partitioned into base and skill groups; (b) four-dimensional cognitive skill representation where the skill bank, composed of general and task-specific skills, is decomposed into four dimensions (D-PLAN, D-TRACK, D-EXEC, D-GUARD) and synthesized back into unified skill descriptions; (c) dimension-focused updates and causal verification where the system focuses on specific dimensions based on agent ability, proposes updates using successful and failed trajectories, and verifies whether to accept or rollback updates.

4.2.2 Causal Verification with Per-Task-Type Rollback

LLM-generated skill updates may incorporate hallucinated facts, introduce conflicting instructions across dimensions, or propose over-specialized rules that contradict existing dimension content, all of which can silently degrade agent performance (Ji et al., 2023). To safeguard against this, we employ a causal verification mechanism with per-task-type rollback that ensures only empirically validated updates are retained.

After each skill evolution step, we save a snapshot of the current skill bank. For each task type m that utilized updated skills during training, we conduct controlled re-evaluation: we regenerate the skill_description from the updated dimensions, inject it into the policy prompt, and re-evaluate by restarting the environment with the same initial states used during training. This causal setup isolates the effect of the skill update from policy learning artifacts. The post-update success rate SR_{post}^m is then compared against the pre-update baseline SR_{skill}^m to determine the true causal impact. A rollback is triggered for task type m only if the performance degrades beyond a tolerance ϵ :

$$\text{Rollback}(m) = \mathbb{1}[SR_{\text{skill}}^m - SR_{\text{post}}^m > \epsilon] \quad (7)$$

The per-task-type granularity is critical: a harmful update to a general skill may degrade performance on one task type while improving another, and a naive rollback mechanism would discard beneficial changes. Our per-task-type approach allows the framework to accept updates that improve the majority of task types while reverting those that cause localized degradation, enabling skill bank evolution to remain robust across task diversity. Additionally, by re-evaluating on the same initial states rather than sampling new rollouts, we minimize variance and ensure that performance differences reflect genuine skill quality.

4.2.3 Competence-Aligned Skill Reward

A critical optimization misalignment in early training prevents the agent from acquiring skill utilization: when the skill group success rate lags behind the base group, GRPO assigns consistently negative advantages to skill-injected trajectories, optimizing the policy to disregard skill prompts. To address this, we propose the **Competence-Aligned Skill Reward (CASR)** that dynamically shapes the reward distribution based on the current skill effectiveness.

Leveraging the skill advantage Δ_m and group baseline μ_m defined in §3.1, the adaptive positive

coefficient λ_m^+ is defined as:

$$\lambda_m^+ = \max(\lambda_{\min}, \lambda_0 \cdot \sigma(-k \cdot \Delta_m)) \quad (8)$$

where λ_0 is the initial amplification factor, k is the decay rate, $\sigma(\cdot)$ is the sigmoid function, and $\lambda_{\min}$ is the lower bound. Intuitively, λ_m^+ is maximized when skills are ineffective and the skill group significantly underperforms, and naturally decays toward $\lambda_{\min}$ as skills become beneficial and the skill group overtakes the base group.

For a trajectory i in the skill group with reward R_i , the deviation from the baseline is $e_i = R_i - \mu_m$. The intrinsic reward is computed asymmetrically:

$$R_i^{\text{int}} = \lambda_m^+ \cdot \max(0, e_i) - \lambda^- \cdot \max(0, -e_i) \quad (9)$$

where λ^- is a fixed small coefficient softening failure penalties. The final shaped reward is $\tilde{R}_i = R_i + R_i^{\text{int}}$ for skill group trajectories, while base group rewards remain unchanged. The shaped rewards $\{\tilde{R}_j\}$ are then fed into the standard GRPO advantage normalization and policy optimization.

CASR exhibits adaptive behavior that evolves with training. In the early stage where skills are ineffective, λ_m^+ stays near its maximum to strongly amplify sporadic successes while λ^- softly penalizes failures, encouraging the policy to explore skill usage. As training progresses and skills become partially effective, λ_m^+ decays proportionally to provide moderate amplification. Once the skill group reliably outperforms the base group, λ_m^+ approaches $\lambda_{\min}$, naturally phasing out reward intervention and restoring standard GRPO optimization.

4.3 Overall Training Pipeline

The complete A²SE training procedure (Algorithm 1, Appendix A) proceeds in two phases. Phase 1 collects initial rollouts and synthesizes the skill bank with four-dimensional structuring. In Phase 2, at each training step, rollouts are partitioned into skill and base groups via A/B masking; CASR shapes the episodic rewards to mitigate the early-stage optimization misalignment before the policy is updated via GRPO. Periodically, the skill bank undergoes ability-aligned evolution: the agent’s ability $A(t)$ is estimated and mapped to the focus dimension set $\mathcal{F}(t)$ to constrain which dimensions the teacher LLM may update, followed by causal verification with per-task-type rollback that retains only empirically beneficial updates.

5 Experiments

5.1 Experimental Setup

We evaluate on ALFWorld (Shridhar et al., 2020), a text-based household environment requiring long-horizon task completion under partial observability, and WebShop (Yao et al., 2022), a realistic web shopping environment where agents search and purchase items matching user specifications. Baselines span four categories: closed-source LLMs (OpenAI O3 (OpenAI, 2025), Claude-Sonnet-4.6 (Anthropic, 2026)), GRPO (Shao et al., 2024) as an RL-only baseline, memory-augmented RL (Mem0+GRPO (Chhikara et al., 2025), SimpleMem+GRPO (Liu et al., 2026)) coupling persistent retrieval with policy optimization, and skill-augmented RL (SkillRL (Xia et al., 2026), D2Skill (Tu et al., 2026)) as our most direct comparisons. We use Qwen2.5-7B-Instruct (Bai et al., 2023) and Qwen3-4B-Instruct-2507 (Yang et al., 2025a) as base models with OpenAI O3 as the teacher LLM, and train purely via RL without cold-start SFT. Full hyperparameters are provided in Appendix C.

5.2 Main Results

A²SE achieves the highest overall success rate across all trainable methods on both benchmarks (Table 1). On ALFWorld with Qwen2.5-7B, A²SE attains 88.3%, surpassing SkillRL by 11.0 and GRPO by 13.3 points, and even exceeding Claude-Sonnet-4.6 (85.9%). On WebShop, A²SE reaches 79.7% success and 91.2 score, outperforming GRPO by 7.1 and 5.2 points. Gains are most pronounced in challenging subtasks, confirming that ability-aligned evolution excels where prior methods fail to inject timely, verified knowledge.

On the smaller Qwen3-4B, A²SE again leads overall at 73.4% (+0.7 over D2Skill, +6.2 over SkillRL), with notable gains on Look (+12.9) and Heat (+14.2). The Clean subtask exception (55.6 vs. SkillRL’s 92.3) suggests that smaller models may lack capacity to fully exploit dimension-targeted updates for certain procedural patterns. Both memory-augmented baselines underperform GRPO (54.7%, 62.5% vs. 75.0%), as raw memory retrieval injects unfiltered context that overwhelms long-horizon decision-making—highlighting the advantage of structured skill abstraction over episodic memory.

Method	ALFWorld							WebShop	
	Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Success
<i>Closed-source LLMs</i>									
O3	64.3	19.1	23.1	64.3	33.3	61.9	43.8	5.8	4.7
Claude-Sonnet-4.6	100	89.3	91.7	90.9	87.5	62.9	85.9	21.1	29.7
<i>Base Model: Qwen2.5-7B-Instruct</i>									
Origin	17.9	4.8	3.8	64.3	0.0	5.3	12.5	16.6	3.9
GRPO	88.3	73.3	<u>76.0</u>	83.3	<u>81.3</u>	40.0	75.0	86.0	<u>72.6</u>
Mem0+GRPO*	78.1	56.1	65.0	54.8	31.0	26.9	54.7	58.1	37.5
SimpleMem+GRPO*	89.5	60.0	64.9	36.3	50.0	26.3	62.5	67.8	46.9
SkillRL	<u>92.3</u>	<u>93.3</u>	58.3	<u>65.0</u>	77.5	<u>51.5</u>	<u>77.3</u>	<u>86.3</u>	71.1
A ² SE	96.9	95.8	77.5	83.3	92.9	75.6	88.3	91.2	79.7
<i>Base Model: Qwen3-4B-Instruct-2507</i>									
Origin	50.0	9.5	0.0	2.1	11.1	4.8	17.2	–	–
GRPO	73.5	46.6	48.0	61.1	<u>62.5</u>	20.0	53.9	–	–
SkillRL*	<u>90.0</u>	92.3	52.0	<u>63.6</u>	42.9	40.9	67.2	–	–
D2Skill*	89.4	<u>72.4</u>	66.7	54.5	60.0	50.0	<u>72.7</u>	–	–
A ² SE	92.3	55.6	<u>56.5</u>	76.5	74.2	<u>44.4</u>	73.4	–	–

Table 1: Main results on ALFWorld and WebShop across different base models. **Bold** and underline denote the best and second-best results among trainable methods, respectively. * indicates results reported from (Tu et al., 2026).

Method	ALFWorld	WebShop
A ² SE (full)	88.3	79.7
<i>Skill Library</i>		
w/o Four-Dim (flat skills)	79.7	68.8
w/o Skill Library (Raw Trj.)	60.9	50.8
<i>Training Pipeline</i>		
w/o Stage-Aware Focus	82.0	72.7
w/o Causal Verification	81.3	62.5
w/o CASR	84.4	75.8
w/o Dynamic Evolution	74.2	65.5

Table 2: Ablation results (success rate %) on ALFWorld and WebShop, using Qwen2.5-7B-Instruct.

5.3 Ablation Study

Having established the overall effectiveness of A²SE, we investigate whether each proposed component is necessary. Table 2 decomposes the framework from two complementary perspectives.

Skill Library Structure. Replacing the four-dimensional representation with flat skill text causes substantial degradation (−8.6 on ALFWorld, −10.9 on WebShop), confirming that dimension-level decomposition is the structural linchpin: without it, neither stage-aware focus nor causal verification can operate, as failures cannot be attributed to specific cognitive dimensions. Re-

moving the skill library entirely in favor of raw trajectory retrieval yields a catastrophic further drop (−27.4, −28.9), underscoring that structured abstraction is indispensable for distilling noisy episodic memory into actionable knowledge.

Training Pipeline Components. Among the pipeline ablations, disabling Causal Verification causes the largest drop on WebShop (−17.2), where hallucinated purchasing advice actively misleads the policy rather than merely being unhelpful. Disabling Dynamic Evolution produces the steepest decline on ALFWorld (−14.1), as static skills cannot adapt to shifting failure modes. Removing Stage-Aware Focus reduces success by 6.3–7.0 points, validating that indiscriminate dimension updates waste the evolution budget and inject premature knowledge. Finally, removing CASR yields a 3.9-point decline on ALFWorld, confirming that without reward shaping, the policy fails to effectively manifest its underlying skills, as the lack of guidance prevents the model from consolidating and utilizing its learned competencies.

5.4 Additional Analysis

5.4.1 Effect of CASR on Training Dynamics

The ablation above confirms that CASR is necessary; we examine *how* it reshapes optimization

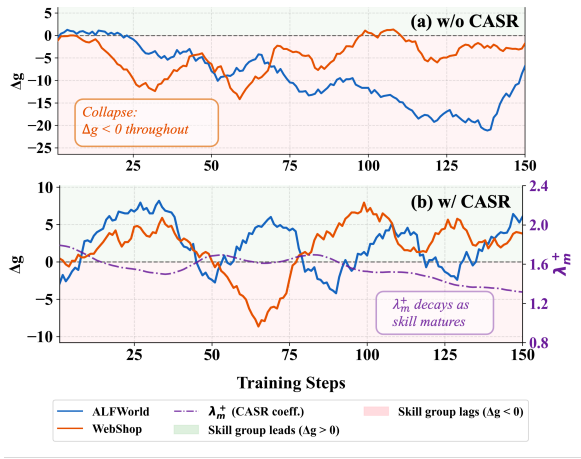


Figure 3: Effect of CASR on training dynamics. $\Delta g = \text{SR}_{\text{skill}} - \text{SR}_{\text{base}}$ throughout training, with and without CASR on ALFWorld and WebShop. The right axis shows the adaptive coefficient λ_m^+ .

dynamics. Figure 3 tracks the gap $\Delta = \text{SR}_{\text{skill}} - \text{SR}_{\text{base}}$ throughout training. Without CASR, Δ remains persistently negative (8–12 pp below zero) during early training; this negative advantage systematically discourages the policy from utilizing skill prompts, effectively muting their contribution to performance. With CASR, the adaptive coefficient λ_m^+ is maximized when the skill group underperforms, amplifying sporadic successes to break the negative feedback loop. The transient dip at steps 50–75 reflects causal-verification latency, after which rollback restores performance. As skills mature, λ_m^+ naturally decays toward $\lambda_{\min}$, seamlessly restoring standard GRPO optimization.

5.4.2 Effectiveness of Causal Verification

While the ablation demonstrates the necessity of causal verification, we further characterize *when and why* rollbacks occur. We quantify the prevalence of harmful skill updates by measuring rollback frequency under causal verification (Figure 4). Across all evolution steps, approximately 34% of per-task-type updates are rolled back—meaning the teacher-LLM-generated content caused measurable degradation in the corresponding task-type success rate beyond tolerance ϵ . This non-trivial rollback rate demonstrates that even strong teacher models regularly produce skills that harm downstream performance, motivating the necessity of empirical verification rather than blind trust in generated content. Crucially, per-task-type granularity enables selective rollback: updates beneficial for some task types are retained while locally harmful

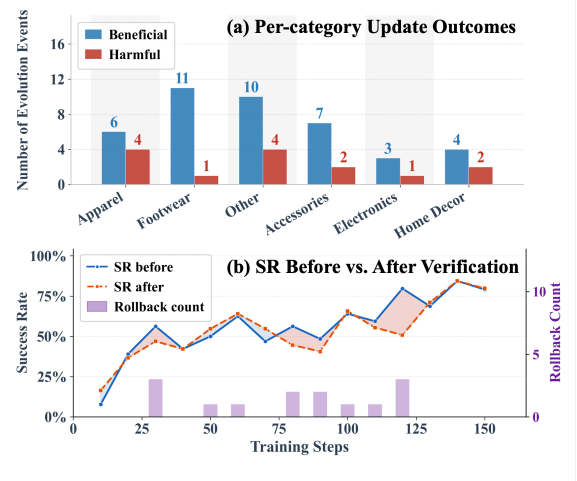


Figure 4: Causal verification statistics on WebShop. (a) Per-category update outcomes: blue bars show retained updates (SR improved or unchanged); red bars show rolled-back updates (SR drop $> 5\%$), with counts annotated above each bar. (b) Overall validation SR before vs. after verification at each evolution step, overlaid with rollback counts (right axis).

ones are reverted, achieving fine-grained risk isolation that a global accept-or-reject strategy would sacrifice.

These analyses are further extended in Appendix B: SFT-initialized experiments validate that A²SE’s gains stem from the evolution mechanism rather than initialization, stage-wise dimension focus visualization traces the ability-aligned transition underlying the ablation penalty, and concrete skill bank examples illustrate both accepted and rolled-back updates in practice.

6 Conclusion

We presented A²SE, an ability-aligned skill evolution framework that resolves two fundamental misalignments in skill-augmented RL. A structured cognitive representation enables precise failure attribution; stage-aware dimension focus and causal verification ensure that only empirically validated, bottleneck-targeted updates enter the skill bank; and the competence-aligned skill reward prevents the policy from prematurely discarding skill prompts during early exploration. Experiments on ALFWorld and WebShop demonstrate consistent improvements over the strongest baselines across different backbone models, with ablation studies confirming each component’s contribution.

Limitations

A²SE depends on a teacher LLM for skill distillation and evolution, making the framework susceptible to the teacher’s knowledge ceiling and generation costs. Moreover, our experiments are limited to text-based benchmarks with constrained action spaces; extending A²SE to multimodal agent settings—where skills must integrate visual, auditory, and motor modalities—poses additional challenges in both representation and verification that remain unexplored.

References

- Huan ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, and 8 others. 2026. [A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence](#). *Preprint*, arXiv:2507.21046.
- Anthropic. 2026. [Introducing claude sonnet 4.6](#).
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. [Qwen technical report](#). *Preprint*, arXiv:2309.16609.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhua Zheng, and Gang Wang. 2026. Multi-agent embodied ai: Advances and future directions. *Science China Information Sciences*, 69(5):151202.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.
- W. Bradley Knox and Peter Stone. 2012. Reinforcement learning from simultaneous human and mdp reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS ’12, page 475–482, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- David R Krathwohl. 2002. A revision of bloom’s taxonomy: An overview. *Theory into practice*, 41(4):212–218.
- Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Binxu Li, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, Yipeng Gao, and 22 others. 2026a. [Skillsbench: Benchmarking how well agent skills work across diverse tasks](#). *Preprint*, arXiv:2602.12670.
- Yu Li, Rui Miao, Zhengling Qi, and Tian Lan. 2026b. [Arise: Agent reasoning with intrinsic skill evolution in hierarchical reinforcement learning](#). *Preprint*, arXiv:2603.16060.
- Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*.
- Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*.
- OpenAI. 2025. [Introducing o3 and o4-mini](#).
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2020. [Alfworld: Aligning text and embodied environments for interactive learning](#). *CoRR*, abs/2010.03768.
- Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. Zerossearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*.

- Songjun Tu, Chengdong Xu, Qichao Zhang, Yaocheng Zhang, Xiangyuan Lan, Linjing Li, Dong Li, and Dongbin Zhao. 2026. Dynamic dual-granularity skill bank for agentic rl. *arXiv preprint arXiv:2603.28716*.
- Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, and Shumin Deng. 2026a. [Skillx: Automatically constructing skill knowledge bases for agents](#). *Preprint*, arXiv:2604.04804.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.
- Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. 2026b. [Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents](#). *Preprint*, arXiv:2604.10674.
- Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. 2026c. Openclaw-rl: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*.
- Tianxin Wei, Ting-Wei Li, Zhining Liu, Xuying Ning, Ze Yang, Jiaru Zou, Zhichen Zeng, Ruizhong Qiu, Xiao Lin, Dongqi Fu, Zihao Li, Mengting Ai, Duo Zhou, Wenxuan Bao, Yunzhe Li, Gaotang Li, Cheng Qian, Yu Wang, Xiangru Tang, and 10 others. 2026. [Agentic reasoning for large language models](#). *Preprint*, arXiv:2601.12538.
- Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xueming Yang, Yufan Shen, Yuxin Wang, and Botian Shi. 2026. [Evolver: Self-evolving llm agents through an experience-driven lifecycle](#). *Preprint*, arXiv:2510.16079.
- Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. [Skillrl: Evolving agents via recursive skill-augmented reinforcement learning](#). *Preprint*, arXiv:2602.08234.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujiu Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Cheng Yang, Xueming Yang, Licheng Wen, Daocheng Fu, Jianbiao Mei, Rong Wu, Pinlong Cai, Yufan Shen, Nianchen Deng, Botian Shi, Yu Qiao, and Haifeng Li. 2025b. [Learning on the job: An experience-driven self-evolving agent for long-horizon tasks](#). *Preprint*, arXiv:2510.08002.
- 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*, 35:20744–20757.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.
- Tik Yu Yim, Wenting Tan, Sum Yee Chan, Tak-Wah Lam, and Siu Ming Yiu. 2026. Asda: Automated skill distillation and adaptation for financial reasoning. *arXiv preprint arXiv:2603.16112*.
- Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2026. [Memagent: Reshaping long-context llm with multi-conv rl-based memory agent](#). In *International Conference on Learning Representations*, volume 2026, pages 39458–39486.
- Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, and Muning Wen. 2026. [Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory](#). *Preprint*, arXiv:2601.03192.
- 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*.
- Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. 2025. [StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21805–21830, Suzhou, China. Association for Computational Linguistics.
- Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. 2025. [Memento: Fine-tuning llm agents without fine-tuning llms](#). *Preprint*, arXiv:2508.16153.
- Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. 2025. [Where llm agents fail and how they can learn from failures](#). *Preprint*, arXiv:2509.25370.

A Algorithm

Algorithm 1 presents the complete training procedure of A²SE, which proceeds in two phases.

Phase 1 (Lines 1–3) initializes the skill bank by first collecting success and failure trajectories from the untrained policy, then prompting the teacher LLM to distill both general and task-specific skills with four-dimensional structuring (D-PLAN, D-TRACK, D-EXEC, D-GUARD).

Phase 2 (Lines 4–16) performs RL with ability-aligned skill evolution. At each training step, tasks are sampled with A/B masking (Line 7) to partition rollouts into a skill group and a base group for computing group-level success rates. The episodic rewards are then shaped by CASR (Line 9), which asymmetrically amplifies positive deviations and softens penalties for the skill group when skills are ineffective, and gradually phases out as the skill group overtakes the base group. The policy is updated via GRPO with the shaped rewards (Line 10). Every Δ_{update} steps, the skill bank undergoes ability-aligned evolution (Lines 11–16): the agent’s current ability $A(t)$ is estimated from group success rates, and mapped to a focus dimension set $\mathcal{F}(t)$ that constrains which cognitive dimensions the teacher LLM may update. Optionally, skills are pruned when the base group exceeds a competence threshold θ_{prune} . Finally, causal verification re-evaluates the updated skill bank per task type and rolls back any changes that degrade performance beyond a tolerance ϵ , ensuring only empirically beneficial updates are retained.

B Extended Results

B.1 Results with SFT Initialization

To examine how different skill-augmented methods benefit from supervised warm-starting, we compare all methods when initialized from the same SFT checkpoint (trained on data from Xia et al. 2026). As shown in Table 3, A²SE achieves the highest overall success rate on ALFWorld (92.2%), outperforming both SkillRL-SFT (89.1%) and D2Skill-SFT (87.8%). On WebShop, A²SE attains 80.5% success rate, comparable to D2Skill-SFT (84.4%). The subtask-level results reveal complementary strengths: D2Skill-SFT excels on procedural execution subtasks (Cool: 95.5%, Heat: 95.0%), while A²SE substantially outperforms on the most challenging Pick2 subtask (87.0% vs. 72.0% and 80.0%). This pattern suggests that D2Skill’s step-level skill design effectively instills fine-grained ac-

tion norms through SFT demonstrations, whereas A²SE’s ability-aligned evolution is particularly advantageous for higher-order cognitive skills where rigid imitation falls short. Overall, even under identical SFT initialization, A²SE’s ability-aligned design yields the best aggregate performance, confirming that the gains stem from the evolution mechanism rather than initialization strategy.

B.2 Stage-Wise Dimension Focus Transition

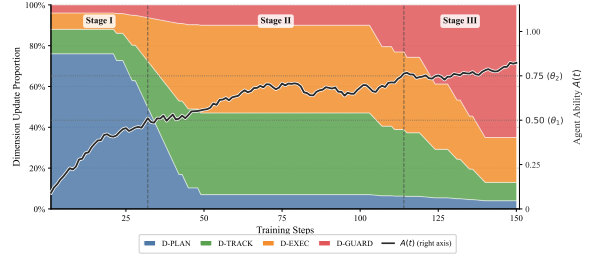


Figure 5: **Stage-wise dimension focus transition.** The stacked area shows the proportion of skill updates allocated to each cognitive dimension across training. Agent ability $A(t)$ (right axis, black line) determines the active stage: D-PLAN dominates in Stage I ($A(t) < \theta_1$), D-TRACK/D-EXEC in Stage II ($\theta_1 \leq A(t) < \theta_2$), and D-EXEC/D-GUARD in Stage III ($A(t) \geq \theta_2$), reflecting the systematic shift of failure modes as training progresses. Dimension proportions are derived from the implemented `_get_focus_dims` schedule ($\theta_1=0.50$, $\theta_2=0.75$); $A(t)$ is computed from real ALFWorld training logs.

The stage-aware dimension focus mechanism dynamically re-weights skill evolution according to the agent’s current ability $A(t)$ (Figure 5). In early training ($A(t) < \theta_1$), agents lack basic exploration strategies and evolution concentrates on D-PLAN, supplying goal-decomposition heuristics. As ability improves into the mid-stage ($\theta_1 \leq A(t) < \theta_2$), failure modes shift from macro-level planning errors to state-loss and execution mistakes; focus correspondingly transitions to D-TRACK and D-EXEC. In late training ($A(t) \geq \theta_2$), failures become concentrated on fine-grained constraint violations, and focus narrows to D-EXEC and D-GUARD. This progression mirrors established theories of staged skill acquisition in cognitive science and directly explains the performance penalty observed when all dimensions are updated uniformly (w/o Stage-Aware): misaligned injection of late-stage knowledge during early exploration dilutes the critical guidance the agent needs most.

Algorithm 1 A²SE: Ability-Aligned Skill Evolution

Require: Base model π_θ , teacher LLM $\mathcal{M}_T$, environment $\mathcal{E}$, skill bank $\mathcal{B}$

Ensure: Trained policy π_{θ^*} , evolved skill bank $\mathcal{B}^*$

// Phase 1: Skill Bank Initialization

- 1: $\mathcal{T}^+, \mathcal{T}^- \leftarrow \text{Rollout}(\pi_\theta, \mathcal{E})$
- 2: $\mathcal{B}_{\text{gen}}, \mathcal{B}_{\text{spec}} \leftarrow \mathcal{M}_T(\mathcal{T}^+ \cup \mathcal{T}^-)$ with four-dimensional structuring
- 3: $\mathcal{B} \leftarrow \mathcal{B}_{\text{gen}} \cup \mathcal{B}_{\text{spec}}$

// Phase 2: RL with Ability-Aligned Skill Evolution

- 4: **for** step $t = 1$ to T_{max} **do**
 - 5: **for** each task m in batch **do**
 - 6: $\mathcal{S}_{\text{ret}} \leftarrow \text{Retrieve}(q_m, \mathcal{B})$
 - 7: Sample $\{\tau_{m,j}\}_{j=1}^N \sim \pi_\theta(\cdot | q_m, \mathcal{S}_{\text{ret}})$ with A/B masking
 - 8: **end for**
 - 9: Compute $\{R_j\}$ via environment; apply CASR to obtain $\{\tilde{R}_j\}$ // Eq. 9
 - 10: Update π_θ via GRPO with shaped rewards $\{\tilde{R}_j\}$
 - 11: **if** $t \bmod \Delta_{\text{update}} = 0$ **then**
 - 12: $A(t) \leftarrow \text{EstimateAbility}(\text{SR}_{\text{skill}}, \text{SR}_{\text{base}})$ // Eq. 5
 - 13: $\mathcal{F}(t) \leftarrow \text{MapFocus}(A(t))$ // Eq. 6
 - 14: **if** $\text{SR}_{\text{base}} > \theta_{\text{prune}}$ **then** Prune($\mathcal{B}$)
 - 15: $\mathcal{B}' \leftarrow \mathcal{M}_T(\mathcal{T}^+, \mathcal{T}^-, \mathcal{F}(t))$
 - 16: $\mathcal{B} \leftarrow \text{CausalVerify}(\mathcal{B}, \mathcal{B}')$ // Eq. 7
 - 17: **end if**
 - 18: **end for**
 - 19: **return** $\pi_\theta, \mathcal{B}$
-

B.3 Generalization to Complex Tasks

To assess whether our approach scales beyond controlled environments, we further evaluate it on **search-augmented question answering**, a setting that demands multi-hop reasoning over potentially noisy retrieval results. We compare A²SE with Search-R1 (Jin et al., 2025), ZeroSearch (Sun et al., 2025), StepSearch (Zheng et al., 2025), EvolveR (Wu et al., 2026) and SkillRL (Xia et al., 2026). A²SE is trained on NQ and HotpotQA and tested on a comprehensive suite of six benchmarks using Qwen2.5-7B as the backbone.

As shown in Table 4, A²SE achieves the highest average score (**44.9**), surpassing all existing RAG agents and skill-augmented baselines. A particularly striking result is its strong zero-shot generalization to unseen multi-hop benchmarks—most notably 2Wiki, where it reaches **45.5**, a substantial margin over the previous best of 42.0 (EvolveR). This confirms that ability-aligned evolution encourages the model to acquire transferable reasoning strategies rather than overfitting to dataset-specific patterns.

B.4 Illustration of Skill Bank

Figure 6(a) presents two representative entries from the initial skill bank. **gen_003** is a general skill applicable across all task types that guides the agent to deliver a held object before resuming exploration; it activates D-PLAN, D-TRACK, and D-GUARD, while D-EXEC remains null since no special execution format is needed. **coo_002** is a task-specific skill for the cool task; it activates D-EXEC, D-TRACK, and D-GUARD, as the primary challenges involve precise operation sequencing and preventing premature placement. Both entries conclude with a `skill_description` field, automatically synthesized by merging all active dimensions into a unified instruction for prompt injection.

Figure 6(b) illustrates two contrasting evolution outcomes. For **gen_003**, the evolution adds a new D-EXEC dimension specifying that the agent must explicitly go to the destination before attempting put, and updates D-GUARD to add a Nothing happens. recovery rule that re-issues go to upon remote-action failure. For **hea_003**, the proposed update adds a new D-EXEC dimension that attempts to convert the general object-verification principle—already expressed in D-GUARD—into

Method	ALFWorld							WebShop	
	Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Success
<i>Closed-source LLMs</i>									
O3	64.3	19.1	23.1	64.3	33.3	61.9	43.8	5.8	4.7
Claude-Sonnet-4.6	100	89.3	91.7	90.9	87.5	62.9	85.9	21.1	29.7
<i>Base Model: Qwen2.5-7B-Instruct</i>									
Origin	17.9	4.8	3.8	64.3	0.0	5.3	12.5	16.6	3.9
GRPO	88.3	73.3	76.0	<u>83.3</u>	81.3	40.0	75.0	86.0	72.6
Mem0+GRPO	78.1	56.1	65.0	54.8	31.0	26.9	54.7	58.1	37.5
SimpleMem+GRPO	89.5	60.0	64.9	36.3	50.0	26.3	62.5	67.8	46.9
SkillRL*	<u>94.3</u>	90.6	92.0	<u>83.3</u>	<u>93.7</u>	<u>80.0</u>	<u>89.1</u>	85.2	72.7
D2Skill*	93.8	<u>94.7</u>	95.5	77.8	95.0	72.0	87.8	90.1	84.4
A ² SE	96.7	100	<u>94.7</u>	88.9	84.2	87.0	92.2	<u>88.7</u>	<u>80.5</u>

Table 3: Performance on ALFWorld and WebShop with SFT-initialized models. All trainable methods are initialized from the same SFT checkpoint trained on data from [Xia et al. 2026](#). **Bold** and underline denote the best and second-best results among trainable methods, respectively. * indicates results reported from ([Tu et al., 2026](#)).

Method	Single-Hop QA			Multi-Hop QA			Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2Wiki [*]	MuSiQue [*]	
<i>Base Model: Qwen2.5-7B-Instruct</i>							
Origin	11.6	35.6	1.20	16.4	22.2	4.80	15.3
Search-R1	39.3	61.0	39.7	37.0	40.1	14.6	38.6
ZeroSearch	43.6	61.8	51.5	34.6	35.2	18.4	40.9
StepSearch	37.7	54.7	38.6	37.0	40.9	34.6	40.6
EvolveR	43.5	63.4	44.6	38.2	42.0	15.6	41.2
SkillRL	45.9	63.3	45.9	43.2	40.3	20.2	43.1
A ² SE	46.1	63.8	45.1	45.3	45.5	23.8	44.9

Table 4: Comparison with state-of-the-art RAG agents and skill-augmented baselines on search-augmented QA benchmarks. †: In-domain; *: Out-of-domain.

a concrete execution rule by hardcoding a specific instance (cup vs. mug). This over-specificity proved harmful: the instruction to “ignore the mismatched item and continue searching” causes the agent to prematurely abandon locations that may simultaneously contain the correct object, directly contradicting D-GUARD’s guidance to examine ambiguous candidates. Causal verification detected this degradation ($\Delta\text{SR}_{\text{heat}} = -0.250$) and reverted the update, underscoring that even plausible-sounding execution rules can conflict with existing dimension content and harm performance.

C Experimental Details

Table 5 summarizes the hyperparameters used in our experiments. The same set of skill evolution and CASR parameters is shared across both bench-

marks unless otherwise noted.

D Prompts

We report the key prompts used in A²SE, organized into two groups: (1) the agent policy prompt fed to the training model at each decision step, and (2) the skill update prompt sent to an external LLM to propose dimension-level revisions.

Agent Policy Prompt. Figure 7 shows the agent prompt template for ALFWorld with skill augmentation. Retrieved skills are prepended under the ##Retrieved Relevant Experience heading; when no skill is retrieved, this block is omitted entirely. The agent outputs a chain-of-thought rationale within tags followed by the selected action within `<action>...</action>` tags. The WebShop prompt follows the same structure with

(a) Representative Skill Bank Entries

gen_003 Deliver Before Resuming Search *When:* Holding any goal object when its target receptacle is known.

D-PLAN Once holding a goal object with an identified destination, delivery is the sole active sub-goal. Navigate directly to the target receptacle and execute put before considering any other action.
D-TRACK Confirm the object is in hand and the destination is identified. After placement, update the delivered count and determine whether more objects are needed.
D-GUARD Do not detour to pick up additional items while carrying an undelivered goal object—this risks missing the delivery window.

skill_description: Once holding a goal object with an identified destination, make delivery the sole active sub-goal and navigate directly to the target receptacle without detours. Issue put immediately upon arrival; confirm placement before any further action. After placement, update the delivered count and determine whether more objects are needed. Do not pick up additional items while carrying an undelivered goal object.

coo_002 Fridge Operation Sequence *When:* Upon arriving at the fridge with the target object in hand.

D-EXEC At the fridge: (1) issue open fridge, (2) issue put [object] in fridge, (3) wait one step or issue the cool action if required, (4) issue take [object] to retrieve the cooled item. Confirm the object is now cooled before leaving.
D-TRACK Track the object’s temperature state. Only advance to the placement phase after cooled is confirmed in observation feedback.
D-GUARD Never place the object at the goal location before retrieving it from the fridge and confirming it is cooled.

skill_description: At the fridge, execute the operation in strict order: open the fridge door, place the object inside, execute the cooling step, confirm the cooled state from observation feedback, then take the object back. Track the object’s temperature state and only advance to the placement phase after cooled is confirmed. Never place the object at the goal location before retrieving it from the fridge and confirming it is cooled. After retrieving the cooled object, navigate directly to the placement destination without detours.

(b) Skill Evolution Cases

gen_003 · Deliver Before Resuming Search Updated dims: D-EXEC (added), D-GUARD

Dim	Before	After	
D-EXEC	(not present)	(new) Must explicitly go to the destination first—never attempt put/move/take from afar. Upon arrival, the immediate next command MUST be put [object] in [receptacle]. Even if the destination is an appliance, use put—NEVER substitute with heat or hallucinated actions.	✓
D-GUARD	Do not detour to pick up additional items while carrying an undelivered goal object.	Never attempt put from afar. If placement returns Nothing happens., re-issue go to. Upon arrival, place immediately—do not navigate away or issue examine first.	✓

hea_003 · Exact Object Verification Updated dim: D-EXEC (added) **rolled back** ($\Delta SR_{\text{heat}} = -0.250$)

Dim	Before	Proposed (reverted)	
D-GUARD	Never pick up a look-alike or substitute object. Before every take action, verify the object type exactly matches the goal specification; if a mismatch is discovered after pickup, drop the object immediately and resume the search.		
D-EXEC	(not present)	(new) Never substitute objects. If you need a cup but find a mug, do not pick it up. Ignore the mismatched item and continue executing search actions on unexamined receptacles until the exact object string is found.	✗

Figure 6: (a) **Representative skill bank entries.** **gen_003** is a general skill applicable across all task types; **coo_002** is a task-specific skill for the cool task. Each entry consists of structured meta-dimensions (only the relevant subset) and a unified skill_description automatically synthesized from all active dimensions for prompt injection. (b) **Contrasting evolution cases.** ✓ marks accepted updates; ✗ marks a rolled-back proposal. **gen_003 (accepted):** The skill converged on three complementary fixes: an explicit go to-before-put prerequisite added as a new D-EXEC, and a Nothing happens. recovery rule in D-GUARD. **hea_003 (rolled back):** The proposed D-EXEC over-specializes a general verification principle from D-GUARD by hardcoding “ignore and keep searching,” which contradicts D-GUARD’s guidance to examine ambiguous candidates and causes the agent to prematurely abandon viable locations.

Hyperparameter	Value
<i>RL Training</i>	
Learning rate	1×10^{-6}
Batch size M	16
Group size N	8
GRPO clip ratio ϵ	0.2
GRPO KL penalty coefficient δ	0.01
Invalid action penalty coefficient	0.1
Max prompt length	3072 / 6144 [†]
Max response length	512
Max steps per episode T	50 / 15 [†]
Total training epochs	150
<i>Skill Retrieval</i>	
Retrieval mode	Embedding
Top- k general skills	2 / 3 [†]
Top- k task-specific skills	2 / 3 [†]
A/B masking ratio	0.5
<i>Skill Evolution</i>	
Update frequency Δ_{update}	10 epochs
Update thresholds	0.4 / 0 [†]
Ability thresholds θ_1, θ_2	0.5, 0.75
Rollback tolerance ϵ	0.05
Prune threshold θ_{prune}	0.6
Max new skills per update	3
Merge threshold	0.85
Validation episodes	16
Teacher LLM $\mathcal{M}_T$	OpenAI O3
<i>CASR</i>	
Initial amplification λ_0	2.0
Failure penalty λ^-	0.3
Decay rate k	4.0
Lower bound $\lambda_{\min}$	0.0

Table 5: Hyperparameters for A²SE. [†]ALFWorld / WebShop respectively.

environment-specific observation and action formats.

ALFWorld — with retrieved skills

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: {task_description}
Retrieved Relevant Skill
{retrieved_skills}
Current Progress
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and actions you took: {action_history}

You are now at step {current_step} and your current observation is:
{current_observation}

Your admissible actions are:
[{admissible_actions}].

Reason step-by-step within <think> </think> tags.
Choose an admissible action within
<action> </action> tags.

Figure 7: Agent policy prompt for ALFWorld with retrieved skills. Without skill augmentation, the **## Retrieved Relevant Skill** block is omitted. WebShop follows the same structure with environment-specific formatting.

Skill Update Prompt. Figure 8 shows the prompt sent to an external LLM at each skill evolution step. It supplies the four meta-dimension descriptions with the training-stage focus marked, the current content of top-retrieved skills, success reference trajectories, and failed trajectories with the skill IDs used. The LLM performs two-phase analysis and returns a JSON object specifying dimension-level updates and optional new skills.

Skill update prompt

You are improving an agent skill bank based on task trajectories.

You have access to SUCCESSFUL and FAILED trajectories for comparison.

SKILL DIMENSIONS:

- D-PLAN: Overall strategy decomposition and search/exploration heuristics
- D-TRACK: Continuous tracking of object states, positions, and task progress
- D-EXEC: Fine-grained action execution details and operation sequences
- D-GUARD: Explicit constraints against common failure modes and error patterns
- [← FOCUS added to the bottleneck dimension(s) for the current stage]

CURRENT SKILLS:

[{skill_id}] {title} (when: {when_to_apply})
Overall: {skill_description}
[D-PLAN] | [D-TRACK] | [D-EXEC] | [D-GUARD]

SUCCESS REFERENCE:

Success Example 1:

Task: {task}
Episode: {N} steps (SUCCESS)
Full trajectory: {step-by-step action/observation pairs}

FAILED TRAJECTORIES:

Failed Example 1:

Task: {task} | Task Type: {task_type} | Episode: {N} steps (FAILED)
Skills used: [{skill_ids}]
Full trajectory: {step-by-step action/observation pairs}

...

PHASE 1 – ROOT CAUSE ANALYSIS

For EACH failed trajectory:

- a) Compare with SUCCESS reference: what did the successful agent do differently?
- b) Identify the failure dimension:
 - D-PLAN if wrong strategy, search order, or sub-goal sequencing
 - D-TRACK if lost track of object location, state, or progress counter
 - D-EXEC if wrong action sequence, missed step, or wrong appliance
 - D-GUARD if violated a known constraint (premature done, skipped step, etc.)
- c) Which skill (if any) contributed? Which dimension?

PHASE 2 – UPDATE INSTRUCTIONS

For each failed trajectory, identify the skill and dimension that caused the failure and provide improved content.

FOCUS: Concentrate on [{focus_dims}] – the current training-stage bottleneck.
May update other dims if root cause clearly points there.

NEW SKILL CRITERIA – add a new skill ONLY if ALL hold:

1. Failure pattern appears across multiple trajectories in this batch
 2. No existing skill (incl. its D-GUARD) addresses this pattern
 3. Pattern is specific and actionable, not a vague restatement
 4. Cannot fix by updating D-GUARD of an existing skill
- When in doubt: update D-GUARD of the most relevant existing skill.

OUTPUT FORMAT

Return a JSON object with exactly two keys:

- "updates": [{ "skill_id": "...", "dim": "{focus_dim}",
"new_content": "improved 1-3 sentences" }]
- "new_skills": [{ "skill_id": "dyn_NNN", "title": "3-5 words",
"when_to_apply": "one sentence",
"meta": {relevant dims only} }]

Return ONLY the JSON, no markdown, no preamble.

Example: {"updates": [{ "skill_id": "gen_001", "dim": "D-PLAN",
"new_content": "Improved content." }], "new_skills": []}

Figure 8: Skill update prompt.