VERIFY IN-SAMPLE, WITHHOLD, DISTILL: THREE INSIGHTS FOR LLM SKILL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill learning has become a popular approach to improving LLM agents: for each training instance, a model reflects on the policy's prediction against the ground truth and writes one natural-language lesson; these lessons are then consolidated into reusable skills that the policy consults at inference time. Some recent work draws a productive analogy with backpropagation. But the two also differ in fundamental ways, and these differences give rise to three insights. **1)** Verify in-sample: a lesson is worth keeping only if it fixes its own training sample. It is common to verify a lesson on a validation set, but that is an out-of-sample test, measuring generalization rather than whether the lesson repairs the sample it was derived from. Verifying in-sample is critical: unlike a sufficiently small gradient step, which by construction reduces the loss on the sample it is computed from, a lesson offers no such guarantee. A lesson should therefore be iteratively fed back to the policy and revised until it fixes its own training sample, and discarded otherwise. **2)** Skill learning overfits far less than backpropagation. A gradient step fits whatever reduces the loss; an individual lesson can overfit its own sample too, but this overfitting is largely suppressed if the ground truth is withheld: the reflection is told whether the prediction was correct and the direction of its error, never the answer itself. Consolidating lessons into skills further discards whatever instance-specific detail remains. Skills learned this way improve performance even from a handful of samples, without the overfitting that few-shot backpropagation exhibits, and gain more as samples grow. **3)** Lessons, not only skills, are worth distilling into smaller models. Most prior work neglects both; the few recent exceptions internalize only the skills into a smaller policy, leaving the lessons unused. Yet the instance-specific detail that consolidation discards, of no use at inference time, is a denser training signal than the skill itself: a self-generated trace of how the policy erred and corrected itself. We give a simple baseline that distills both. Each insight is simple, yet all three are routinely overlooked; in our experiments, acting on them consistently improves skill learning and surpasses prior state of the art.

1 INTRODUCTION

Skill learning improves an LLM agent without updating its parameters. The skill learning pipeline typically has two stages, *i.e.*, drawing lessons and consolidating lessons into skills. First, for each training sample, a model examines the policy's prediction in light of the ground truth and writes one natural-language *lesson* stating what the policy did right or wrong on that sample (Shinn et al., 2023; Zhao et al., 2024). Second, the lessons are then consolidated into reusable *skills* that the policy reads at inference time (Ni et al., 2026; Zhang et al., 2026a; Yang et al., 2026).

Recent work draws an explicit analogy between skill learning and backpropagation. The idea originates in prompt optimization, where natural-language critiques are treated as textual gradients and applied as updates (Pryzant et al., 2023; Yuksekgonul et al., 2024). SkillOpt (Yang et al., 2026) carries the analogy into skill learning, accepting an update only if the skill scores higher on a validation split, and reports the strongest skill-learning results to date.

Skill learning, however, differs from backpropagation in three fundamental ways, and each difference calls for a change in how skill learning is done. We state the three insights, and test each of them.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

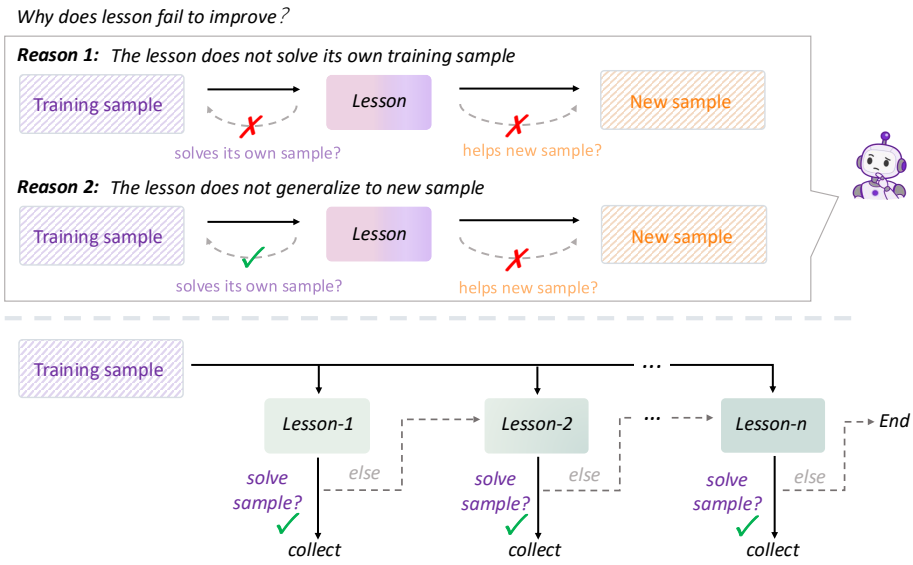


Figure 1: **Top:** *A lesson may fail in two ways*: it does not help the policy solve its own training sample (Reason 1), or it helps solve that sample but fails to generalize to new ones (Reason 2). Prior work focuses primarily on generalization, while source-sample correction receives less attention. **Bottom:** *Iterative revision during drawing lessons.* A lesson is tested by rerunning the policy on its source training sample. If the policy solves the sample, the lesson is collected, otherwise the lesson is revised based on the failed attempt and tested again. This process continues until a lesson leads to success.

Insight 1: Verify in-sample: a lesson is worth keeping only if it fixes its own training sample. A sufficiently small gradient step in backpropagation reduces the loss on its source sample, whereas a lesson offers no corresponding guarantee. A lesson may fail to solve its own training sample or to generalize to new ones (Figure 1, top). Existing pipelines emphasize the latter (Alzubi et al., 2026; Yang et al., 2026), while source-sample correction is often overlooked. **Our response** is *iterative revision* for drawing lessons. Given a training sample, the policy repeatedly revises and tests candidate lessons on that sample until it derives an effective lesson that helps it solve the sample (Figure 1, bottom). Upon success, we draw a lesson from the full failure-to-success trajectory, capturing how the errors were corrected and providing informative experience for consolidation into reusable skills.

Insight 2: Withhold the answer: skill learning then overfits far less than backpropagation. Fitting individual training samples, as Insight 1 requires, would raise concerns about overfitting under backpropagation. The two stages of the skill learning pipeline provide complementary safeguards against this risk: when drawing lessons, withholding the reference answer and providing only diagnostic hints limits direct answer copying; when consolidating lessons into skills, lessons are abstracted into reusable procedures shared across samples, reducing reliance on instance-specific details.**Our takeaway** is that, as long as these safeguards are kept in place, the iterative revision of Insight 1 can be pushed to solve each source sample without incurring the overfitting that such per-sample fitting would cause under backpropagation. Two observations support this claim. First, as the cap on iterative revisions rises, more training samples are solved, the very regime that would drive overfitting under backpropagation, yet test accuracy improves rather than degrades (Figure 3(a)). Second, in the low-data settings evaluated, skill learning improves test performance from only a handful of samples, precisely where backpropagation overfits most, whereas the SFT baseline degrades (Figure 3(b)). Beyond these popular safeguards, we further introduce additional practice to strengthen resistance to overfitting even more.

Insight 3: Lessons, not only skills, are worth distilling into smaller models. Most prior work does not distill either lessons or skills into smaller models. The few recent distillation efforts use only consolidated skills from a stronger policy, without incorporating the underlying lessons. These lessons preserve a detailed record of how the policy erred and corrected itself, and can therefore provide richer supervision than the abstracted skills alone. **Our exploration** of this insight results in

a simple distillation baseline. It follows a standard training pipeline of supervised fine-tuning (SFT) followed by reinforcement learning (RL). Its key feature is that it distills not only consolidated skills but also the underlying lessons from a stronger policy. This simple baseline achieves substantial performance gains in the smaller model.

We summarize our expressions of these three insights as follows. Insight 1 calls for iterative revision during drawing lessons: each lesson is tested on its source sample and revised after failed attempts, with only lessons that lead to successful correction retained for consolidation. Insight 2 allows iterative revision with limited risk of overfitting, provided that the safeguards at both stages are maintained in skill learning pipeline. Insight 3 motivates a simple distillation baseline that combines a stronger teacher's lessons and skills to improve smaller models. Our experiments show that these practices yield substantial performance gains in smaller models.

2 Related Work

Prior work on skill learning. An agent skill is a structured, reusable document that encodes task procedures, domain knowledge, and operational guidelines (Li et al., 2026b; Jiang et al., 2026b), and skill learning writes such a document from the agent's own execution experience by reflection and consolidation. Methods differ in how each stage is realized. Reflection may read a trajectory as a whole (Liu et al., 2026b; Yu et al., 2026), or assign separate analysts to successes and failures, with the failure analyst running as an agentic loop that inspects files and re-examines the trace before writing (Ni et al., 2026). Earlier experiential agents keep the lessons as retrievable memories (Zhao et al., 2024; Ouyang et al., 2026b); later methods merge lessons hierarchically (Ni et al., 2026), edit the skill incrementally so that earlier content survives (Zhang et al., 2026a; Alzubi et al., 2026), or curate lessons into a repository (Jiang et al., 2026a; Wang et al., 2026a; Ouyang et al., 2026a). We do not propose another variant. Our focus is on three practical questions: how to verify and revise lessons on their source samples, how to preserve generalization during this revision, and what to distill from stronger policy into smaller ones.

The backpropagation analogy. Recent work draws an analogy between natural-language feedback and gradient-based learning. The idea originates in prompt optimization, where critiques are applied as updates (Pryzant et al., 2023; Yuksekgonul et al., 2024), and in reflective prompt evolution (Agrawal et al., 2026). Training-Free GRPO replaces the numerical group advantage with a semantic one distilled into an experience library (Cai et al., 2025). In skill learning, SkillOpt (Yang et al., 2026) equips skill optimization with rollout batches, a textual learning rate and schedule, and a held-out acceptance gate. The correspondence is incomplete in one respect: existing pipelines validate an update at the level of the consolidated skill, on a held-out split (Alzubi et al., 2026; Yang et al., 2026), and nothing checks whether a lesson fixes the failure it was written from, which a gradient step is guaranteed to do. We add this check explicitly: the policy writes a lesson on its failure and reruns the sample, revising the lesson until the sample is solved, and a final lesson is written only then.

Transfer and distillation of skills. Because a skill is text rather than a weight update, it is not tied to the model that produced it: skills learned with one model have been shown to improve others across scales and families (Ni et al., 2026). This raises the question of how far the benefit can be moved. One line keeps the recipient frozen and delivers the skill through its context, as in teacher-student transfer where a patch learned from a stronger model is read by a weaker one at inference time (Shi et al., 2026a). Another internalizes skills into the parameters: Skill-SD distills a skill-conditioned teacher into a student acting under the plain prompt (Wang et al., 2026b), SkillRL co-evolves a skill library with the policy under reinforcement learning (Xia et al., 2026), and SkillC and Skill0 study internalization through contrastive credit assignment and in-context agentic reinforcement learning (Lin et al., 2026; Lu et al., 2026; Shi et al., 2026b). None of these fine-tunes the smaller policy on the stronger model's lessons, the record of how it erred and corrected itself. We give a baseline that distills both the lessons and the skill.

3 Skill Learning with Iterative Lesson Revision

Insight 1 suggests repeatedly revising and testing lessons on their source samples, retaining only those that lead to successful correction. Insight 2 suggests that this correction can be pursued more thoroughly with limited risk of overfitting. We therefore augment the standard skill learning pipeline

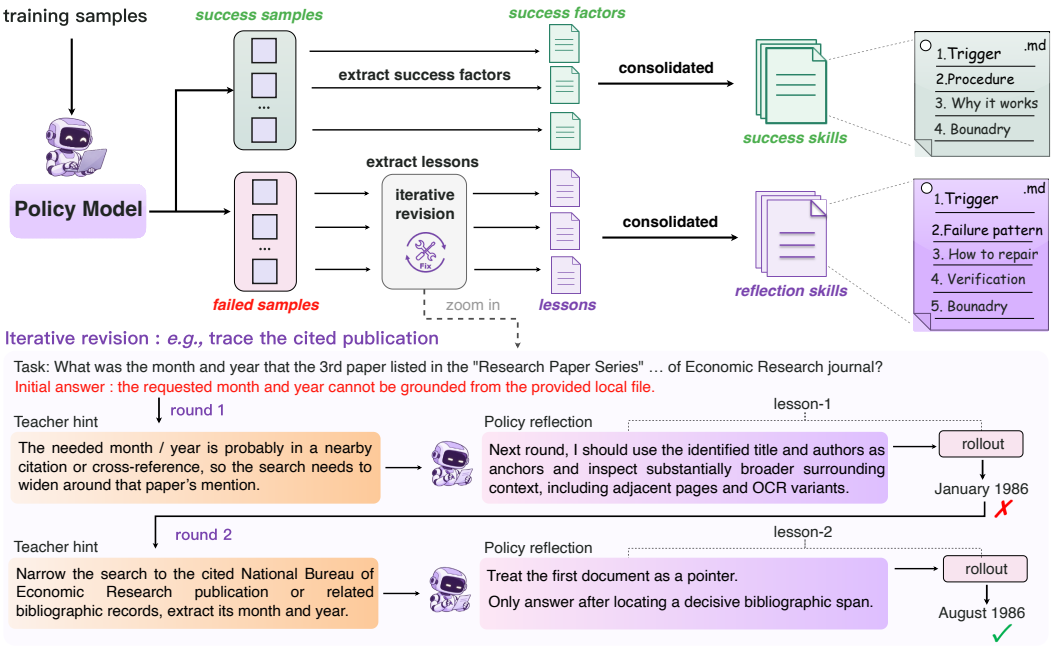


Figure 2: **Top**: *Overall skill learning pipeline*. The policy first attempts each training sample. Initially successful samples yield success factors, while failed samples enter iterative revision. Samples solved within the revision budget yield final lessons summarized from their failure-to-success trajectories. Success factors and final lessons are then consolidated into a success skill and a reflection skill, respectively. **Bottom**: *Iterative revision during drawing lessons*. Given a failed sample, the teacher first provides a diagnostic hint. Guided by this hint, the policy writes a reflection and retries the same sample. The reflection and the resulting policy rollout together constitute one *lesson*. If the retry fails, the teacher provides a new hint on the latest trajectory, and the policy revises its previous reflection before trying again. In this example, the first retry fails, while the second succeeds after the policy redirects its search to the cited publication.

with iterative revision during lesson drawing. Using this pipeline, we empirically assess these two insights (Figure 3).

Overall Skill Learning Pipeline. Our pipeline follows the standard two-stage structure of drawing lessons and consolidating lessons into skills, with iterative revision added to the first stage (Figure 2). The policy first attempts each training sample. An initially successful attempt yields a *success factor* describing the procedure followed and the conditions under which it applies. These samples require no corrective revision because there is no observed failure to repair. Initially failed samples instead enter iterative revision, contributing lessons only if they are solved within the revision budget.

During consolidation, success factors are abstracted into a *success skill*, and lessons from iterative revision into a *reflection skill*. Each success-skill entry specifies when a procedure applies, how it works, why it works, and its limitations. Each reflection-skill entry specifies when an error pattern may arise, how to correct it, how to verify the correction, and the limits of its applicability.

At inference, the policy first attempts a new sample with the success skill in context. It then reviews its attempt using the reflection skill, correcting any identified error or confirming the initial attempt otherwise. Neither the teacher nor the evaluator is used at inference.

Iterative revision during drawing lessons. In each revision round, the teacher examines the latest failed trajectory and provides a diagnostic hint identifying the error and suggesting a direction for correction, without revealing the reference answer. Guided by this hint, the policy writes or revises a reflection specifying how to correct the failure, then retries the same sample with the reflection in context. The reflection and the resulting policy rollout together constitute one *lesson*.

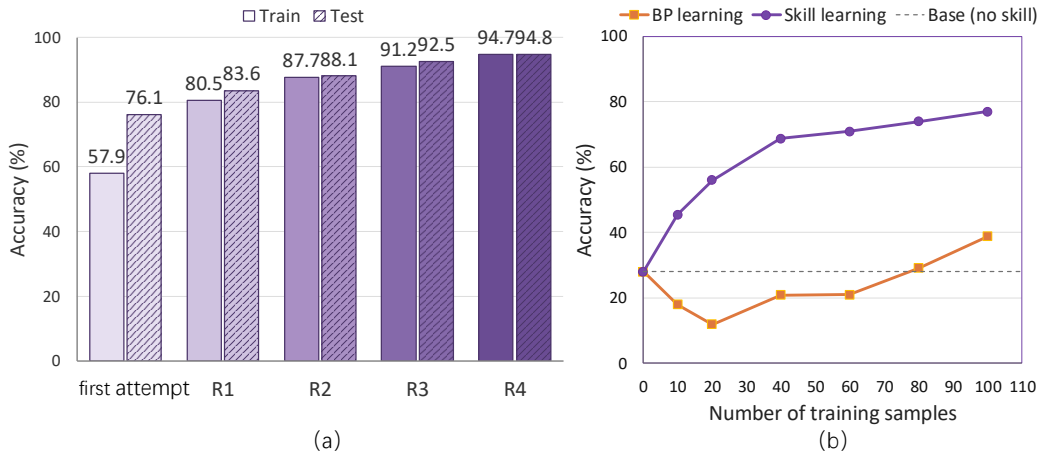


Figure 3: **Exploring iterative revision and generalization on ALFWorld.** (a) Effectiveness of iterative revision. For GPT-5.4, *Train* denotes the fraction of training samples solved within each budget, including retries guided by teacher hints; *Test* denotes test success with the skills learned under that budget, without teacher hints. The budget ranges from zero to four revision rounds (R1–R4). With no revision, test inference uses only the success skill. (b) Overfitting risk with limited training data. Test success of Qwen3-4B using skills learned through iterative revision versus SFT on the same training samples. Training-set sizes range from 10 to 100; the zero-sample point and dashed line indicate the base policy. Skill learning improves on this baseline at every evaluated size from 10 to 100, whereas SFT falls below it when fewer than 80 samples are used.

Figure 2 (bottom) illustrates this process: the first round broadens the search around the paper's mention but returns an incorrect date; the second redirects the search to the cited publication and obtains the correct answer. Revision stops at the first successful retry or after K rounds, with $K = 4$ in our experiments. Upon success, the full failure-to-success trajectory is summarized into a *final lesson* for consolidation.

Discussion the effect of iterative revision. We first examine whether additional revision corrects more source-sample failures and produces more useful skills. For GPT-5.4 on ALFWorld, we vary the revision budget from zero to four rounds, tracking the fraction of training samples solved within the budget and the test performance of the skills learned under each budget. As shown in Figure 3(a), the training success rate increases from 57.9% to 94.7%, while the corresponding test success rate rises from 76.1% to 94.8%. Both improve with additional rounds, supporting the value of continuing iterative revision when drawing lessons.

Discussion the risk of overfitting. We next consider whether repeatedly fitting the source samples harms generalization. The continued test gains in Figure 3(a) suggest that more thorough revision need not come at the expense of generalization. To further explore generalization with limited data, we compare skill learning and SFT for Qwen3-4B on ALFWorld using the same training samples, with training-set sizes ranging from 10 to 100. In Figure 3(b), skill learning improves on the base policy at every size, including with only 10 samples, whereas SFT falls below the base policy when fewer than 80 samples are used. Under the feedback restrictions and consolidation used here, these observations suggest that skill learning can benefit from repeated revision and generalize from small training sets.

4 A Distillation Baseline

Following Insight 3, this section proposes a simple distillation baseline that ultilizes both lessons and skills.

Supervised fine-tuning with lessons. The strong policy runs the training samples by iterative revision as Insight 1 requires, and the sequences it produces form the fine-tuning data: the trajectory of each sample solved at the first attempt, and the lessons of each sample solved after revision, up to

the successful run. The small policy is fine-tuned on this data, so that it fits how the strong policy reasoned, erred, and corrected itself. This is the stage that prior work lacks. After fine-tuning, the small policy runs iterative revision on the same samples and consolidates a reflection skill. In effect this skill derives from the strong policy, as the small policy has been fine-tuned on the strong policy's lessons.

Reinforcement learning with skills. The fine-tuned small policy is then trained by reinforcement learning with correctness as the reward. To distill the reflection skill from the previous stage into its parameters, a failed rollout is retried with the reflection skill in the context, and a successful retry is fit by a cross-entropy loss. Specifically, for a sample x, the policy generates a group of rollouts as in GRPO (Shao et al., 2024). In addition to the standard GRPO update, we retry each failed rollout τ once with the reflection skill in context, obtaining τ', and apply a cross-entropy loss to τ' if it succeeds. This auxiliary loss is computed under the original task context, without the reflection skill, to train the policy to reproduce the successful behavior without relying on the skill at inference. The overall objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{RL}}\big(\theta;\, x, \tau\big) + \mathbb{1}\big[r(\tau) = 0,\; r(\tau') = 1\big]\, \mathcal{L}_{\mathrm{CE}}\big(\theta;\, x, \tau'\big), \tag{1}$$

where $\mathcal{L}_{\mathrm{RL}}$ is the GRPO objective on rollouts without the skill, and r is the correctness label. In this way, the behavior that the reflection skill induced on the retry is written into the parameters rather than left in the context. At inference the small policy runs with no skill in its context.

5 Experiments

Evaluation. We evaluate in the two settings in which skill learning is used. In the first, the policy is frozen and reads the learned skill in its context. We test it on two suites, following the setting of the original work in each: a text-only suite of search, office-document, and document-QA tasks and embodied decision making (SearchQA (Dunn et al., 2017), OfficeQA (Opsahl-Ong et al., 2026), DocVQA (Mathew et al., 2021), ALFWorld (Shridhar et al., 2020)), and a multimodal suite of visual tool use, multimodal search, and long-horizon agent tasks (VisualToolBench (Guo et al., 2025), TIR-Bench (Li et al., 2026a) , MMSearch-Plus (Tao et al., 2026), AgentVista (Su et al., 2026)). In the second setting, a small open-source policy is trained by the distillation baseline of Section 4 and runs with no skill in its context. We test it on ALFWorld and WebShop (Yao et al., 2022a), following the setting of SkillRL (Xia et al., 2026).

Implementation details. The policies are GPT-5.5, GPT-5.4, and GPT-5.4-mini on the first suite and Gemini-2.5-Pro, Gemini-3-Flash, and GPT-5-mini on the second. Each learns its own skill from its own trajectories, and the hints are written by Kimi-K2.6. A failed sample is revised up to four times. Generation parameters follow the original work of each suite. On the second suite we follow the per-benchmark tool sets of XSkill: TIR-Bench exposes only the Python code interpreter; VisualToolBench exposes web search, page visit, and the code interpreter; MMSearch-Plus and AgentVista additionally expose image search. All baselines and our method share these tool sets, and the tool descriptions are the unmodified system prompts released with the benchmark. Search and page retrieval are served by Serper and Jina respectively, and the code interpreter runs in a stateful Python sandbox.

For distillation, the small policy is Qwen3-4B (Yang et al., 2025) or Qwen2.5-7B-Instruct (Qwen et al., 2025), the base model of the baselines in Table 3, and the strong policy is Qwen3-235B, with the hints written by Kimi-K2.6 as above. The strong policy runs iterative revision on ALFWorld and WebShop training games, the small policy is fine-tuned for 8 epochs with learning rate 2e-5 and batch size 8. Reinforcement learning then uses GRPO with learning rate 1e-6, batch size 16 , for 150 training steps. Evaluation follows SkillRL. More details and prompts in Appendix A.

5.1 Evaluation on the skilling learning results

We evaluate skills learned through iterative revision via in-context use on the text and multimodal benchmark suites (Tables 1 and 2). We also evaluate distillation of lessons and skills into smaller models on ALFWorld and WebShop (Table 3). Compared with prior methods, our learned skills yield the highest performance on benchmarks in the two suites, while our distillation baseline achieves the highest average performance on ALFWorld and WebShop. The key observations are as follows.

Table 1: Test success rate (%) on the four benchmarks in the text-only suite. Avg. is the unweighted mean across the four benchmarks. Within each policy, the best result per column is in **bold** and the second-best is underlined.

Model	Skill source	SearchQA	OfficeQA	DocVQA	ALFWorld	Avg.
GPT-5.5	No skill	77.7	33.1	78.8	83.6	68.3
	Human skill	81.8	66.9	90.1	91.8	82.7
	LLM skill	80.9	51.7	89.6	93.3	78.9
	Trace2Skill	82.4	65.7	90.6	87.3	81.5
	TextGrad	81.4	42.0	87.2	82.8	73.4
	GEPA	84.8	63.9	89.1	85.8	80.9
	SkillOpt	87.3	72.1	91.2	95.5	86.5
	Ours	**89.4**	**76.2**	**95.0**	**96.3**	**89.2**
GPT-5.4	No skill	76.9	50.0	77.6	75.4	70.0
	Human skill	80.1	59.3	88.5	74.6	75.6
	LLM skill	79.4	20.4	90.4	83.6	68.5
	Trace2Skill	81.2	51.2	89.3	73.1	73.7
	TextGrad	82.5	54.7	85.3	78.4	75.2
	GEPA	82.4	60.4	88.3	74.6	76.4
	SkillOpt	83.1	62.8	91.2	91.0	82.0
	Ours	**87.5**	**66.9**	**94.9**	**94.8**	**86.0**
GPT-5.4-mini	No skill	75.9	22.1	71.4	73.1	60.6
	Human skill	77.2	45.9	85.0	56.7	66.2
	LLM skill	76.1	36.6	86.4	65.7	66.2
	Trace2Skill	78.6	20.9	88.5	82.8	67.7
	TextGrad	77.5	30.0	84.0	70.9	65.6
	GEPA	79.4	45.3	83.7	81.3	72.4
	SkillOpt	80.2	48.8	90.9	85.8	76.4
	Ours	**83.1**	**51.2**	**93.8**	**91.0**	**79.8**

In-context skill learning on the text benchmarks. As shown in Table 1, using skills learned through iterative revision improves performance over using no skill across all four benchmarks and all three policies. With GPT-5.5, these skills raise ALFWorld success from 83.6% to 96.3% and SearchQA success from 77.7% to 89.4%. With GPT-5.4-mini, the corresponding success rates increase from 73.1% to 91.0% and from 75.9% to 83.1%. The gains are larger on OfficeQA: GPT-5.5 improves from 33.1% to 76.2%, and GPT-5.4-mini from 22.1% to 51.2%.

Compared with SkillOpt (Yang et al., 2026), our learned skills yield success rates that are 0.8 to 5.2 percentage points higher across all evaluated benchmark. The largest margin is on ALFWorld with GPT-5.4-mini, where success increases from 85.8% with SkillOpt to 91.0% with our learned skills.

In-context skill learning on the multimodal benchmarks. The same holds on the multimodal benchmarks (Table 2), where the reference is the tool-using policy and the tasks require image tools and web search. With Gemini-2.5-Pro, ours improves TIR-Bench from 28.38 to 59.00, MMSearch-Plus from 21.56 to 48.34, and VisualToolBench from 25.35 to 47.20; with Gemini-3-Flash, TIR-Bench goes from 32.37 to 63.00 and AgentVista from 20.64 to 33.94; with GPT-5-mini, MMSearch-Plus goes from 14.22 to 24.60. Across the four benchmarks, our pipeline improves average success over the tool-using policy by 20.96, 17.54, and 7.65 percentage points for Gemini-2.5-Pro, Gemini-3-Flash, and GPT-5-mini, respectively. This demonstrates that the gain of iterative revision carries over to tasks that require image tools and web search.

Compared with XSkill (Jiang et al., 2026a), the state-of-the-art method on these benchmarks, ours surpasses it by 16.2 points on average with Gemini-2.5-Pro, 10.8 with Gemini-3-Flash, and 5.1 with GPT-5-mini; XSkill itself exceeds the tool-using policy by 4.8, 6.7, and 2.6. The margin is widest on TIR-Bench, 25.9 points with Gemini-2.5-Pro and 15.3 with Gemini-3-Flash, and on MMSearch-Plus, 20.4 with Gemini-2.5-Pro. XSkill extracts its experiences and skills from rollouts and never reruns the sample they came from; this demonstrates that the gain of iterative revision over accumulation without a check holds on multimodal tasks as on text.

Table 2: Test success rate on the benchmarks of multimodal suite. Best per column in **bold**, second-best underlined.

Model	Method	VisualTool	TIR-Bench	MMSearch-Plus	AgentVista	Avg
Gemini-2.5-Pro	No tools	20.91	21.37	10.43	17.89	17.65
	w/ tools	25.35	28.38	21.56	20.18	23.87
	AWM	25.93	29.75	20.85	19.72	24.06
	DC	24.77	27.62	24.64	21.79	24.71
	Agent-KB	26.75	29.13	23.22	20.87	24.99
	XSkill	30.49	33.12	27.96	22.94	28.63
	Ours	**47.20**	**59.00**	**48.34**	**24.77**	**44.83**
Gemini-3-Flash	No tools	25.12	28.50	16.47	18.35	22.11
	w/ tools	41.94	32.37	39.57	20.64	33.63
	AWM	41.94	34.25	43.36	19.50	34.76
	DC	41.70	33.75	40.28	20.18	33.98
	Agent-KB	41.75	36.62	39.81	21.33	34.88
	XSkill	46.50	47.75	43.72	23.39	40.34
	Ours	**57.01**	**63.00**	**50.71**	**33.94**	**51.17**
GPT-5-mini	No tools	13.90	20.00	3.08	18.58	13.89
	w/ tools	24.30	23.50	14.22	20.41	20.61
	AWM	23.25	24.25	14.81	19.04	20.34
	DC	23.83	24.50	13.74	21.10	20.79
	Agent-KB	24.77	25.13	13.63	19.95	20.87
	XSkill	24.53	28.25	16.11	23.85	23.19
	Ours	**32.24**	**30.50**	**24.60**	**25.69**	**28.26**

Table 3: Success rate on ALFWorld per task type and on WebShop. Ours is the distillation baseline with Qwen3-4B and Qwen2.5-7B-Instruct as the small policy. Best **bold**, second-best underlined.

	ALFWorld							WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg.	Score	Succ.
w/o Training									
ReAct (Yao et al., 2022b)	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Reflexion (Shinn et al., 2023)	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
Mem0 (Chhikara et al., 2025)	54.0	55.0	26.9	36.4	20.8	7.7	33.6	23.9	2.0
ExpeL (Zhao et al., 2024)	21.0	67.0	55.0	52.0	71.0	6.0	46.3	30.9	11.2
RL-Trained w/o Skills									
PPO (Schulman et al., 2017)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	81.4	68.7
RLOO (Ahmadian et al., 2024)	87.6	78.2	87.3	81.3	71.9	48.9	75.5	80.3	65.7
GRPO (Shao et al., 2024)	90.8	66.1	89.3	74.7	72.5	64.7	77.6	79.3	66.1
GiGPO (Feng et al., 2026)	97.7	82.7	98.8	83.7	89.3	79.2	90.8	84.4	72.8
RL-Trained w/ Skills									
EvolveR (Wu et al., 2025)	64.9	33.3	46.4	13.3	33.3	33.3	43.8	42.5	17.6
Mem0 (Chhikara et al., 2025) w/ GRPO	78.1	54.8	56.1	31.0	65.0	26.9	54.7	58.1	37.5
SimpleMem (Liu et al., 2026a) w/ GRPO	89.5	36.3	60.0	50.0	64.9	26.3	62.5	67.8	46.9
SkillRL (Xia et al., 2026)	97.9	71.4	90.0	90.0	**95.5**	87.5	89.9	85.2	72.7
RetroAgent (Zhang et al., 2026b)	97.9	90.9	**99.2**	92.9	85.3	91.0	94.9	88.9	82.3
Ours (Qwen3-4B)	**100.0**	**100.0**	97.1	94.7	91.7	95.8	96.4	90.8	83.6
Ours (Qwen2.5-7B)	**100.0**	**100.0**	97.1	**100.0**	91.7	**100.0**	**97.9**	**92.6**	**85.9**

Distillation of lessons and skills. As shown in Table 3, we report the distilled policy with two small models, Qwen3-4B and Qwen2.5-7B-Instruct, the latter being the base model of the baselines; both use no skill in context at inference. Compared with the methods that train a policy by reinforcement learning without a skill, the 7B policy improves on five of the six ALFWorld task types and reaches 97.9 overall against 90.8 for GiGPO, the strongest of them, with Look rising from 82.7 to 100.0 and Pick2 from 79.2 to 100.0; on WebShop it reaches a success rate of 85.9 against 72.8. The 4B policy

Table 4: Ablation of the two skills on ALFWorld with GPT-5.4-mini, test success rate.

Setting	Success
No skill	73.1
Success skill only	81.3
Reflection skill only	88.8
Both (ours)	91.0

Table 5: Transfer and distillation on ALFWorld

Setting	Success
No skill	28.0
Own skill in context	76.9
Teacher skill in context	81.3
Lessons distilled (fine-tuning only)	87.5
No Lessons distilled (fine-tuning only)	70.9
Lessons and skill distilled (ours)	96.4

reaches 96.4 on ALFWorld and 83.6 on WebShop. This demonstrates that the lessons of a strong policy carry more into a small policy than a task reward alone.

Compared with the state-of-the-art methods that train with skills, RetroAgent and SkillRL , the 7B policy surpasses them on ALFWorld by 3.0 and 8.0 points overall and on WebShop by 3.6 and 13.2 points in success rate; the 4B policy, with fewer parameters than either, surpasses them by 1.5 and 6.5 on ALFWorld and by 1.3 and 10.9 on WebShop. Both are best on four of the six ALFWorld task types, with the 7B policy reaching 100.0 on Pick, Look, Heat, and Pick2, and second on Cool, where SkillRL leads. Ours trains on a strong policy's lessons and uses a skill only to retry failed rollouts; this demonstrates that a recipe built from supervised fine-tuning and reinforcement learning is a competitive baseline for internalizing skills.

5.2 Ablations and analysis

We ablate the two skills on ALFWorld with GPT-5.4-mini (Table 4). Reading the success skill alone raises the policy from 73.1 to 81.3; reading the reflection skill alone improves it to 88.8; reading both gives 91.0. The reflection skill thus carries the larger share of the gain, and the two are complementary. This demonstrates that the lessons learned from revised samples, which record errors and their corrections, contribute more than the lessons from samples solved at once, by letting the policy check its own attempt against the recorded errors.

Analysis of lesson and skill distillation (Insight 3). We evaluate the effectiveness of lesson and skill distillation for Qwen3-4B on ALFWorld, comparing in-context skill use, fine-tuning with and without lessons, and the full distillation baseline (Table 5). Reading its own skill improves the policy's success rate from 28.0% to 76.9%, while reading the stronger policy's skill reaches 81.3%. Fine-tuning on the stronger policy's sequences of lessons reaches 87.5%, compared with 70.9% for fine-tuning without lessons, a gain of 16.6 percentage points. This comparison supports including lessons in the distillation data: the error-correction experience they preserve provides useful supervision for the smaller model. Lesson-based fine-tuning also exceeds in-context use of the policy's own skill and the stronger policy's skill by 10.6 and 6.2 percentage points, respectively. Subsequent reinforcement learning with skill-guided retries further raises the success rate to 96.4%, an additional gain of 8.9 percentage points over lesson-based fine-tuning.

6 Conclusion

This paper presents three insights for LLM skill learning. In-sample verification motivates iterative revision when drawing lessons: the policy revises and retries failed attempts, retaining final lessons only from samples solved within the revision budget. With reference answers withheld during lesson drawing and instance-specific details abstracted during consolidation, this revision can improve source-sample fitting with limited risk of overfitting. Our experiments show that additional revision improves test performance and that skill learning remains effective with few training samples. Beyond their role in consolidation, lessons preserve error-correction experience that provides useful supervision for smaller models. We explore this insight through a simple SFT–RL baseline that learns from a stronger policy's lessons and subsequently internalizes a reflection skill. Across the evaluated text and multimodal benchmarks, our skill learning pipeline outperforms the compared methods. The distillation baseline also yields substantial gains on ALFWorld and WebShop, without requiring skills in context at inference.

AI USE STATEMENT

We used LLMs to assist with language polishing, including grammar correction and improvements to clarity and readability, as well as benchmark evaluation. The research ideas, methodology, experimental design, and interpretation of results remain the responsibility of the authors. All AI-assisted text and evaluation results were reviewed and verified by the authors, who take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

All results reported in our paper are fully reproducible. We describe the skill learning pipeline and iterative revision procedure in section 3, and the distillation baseline in section 4. The experimental setup and evaluation protocol are provided in section 5. To support reproducibility, we provide the prompt templates in Appendix A.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*, 2026.

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, et al. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*, 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Xingang Guo, Utkarsh Tyagi, Advait Gosai, Paula Vergara, Jayeon Park, Ernesto Gabriel Hernández Montoya, Chen Bo Calvin Zhang, Bin Hu, Yunzhong He, Bing Liu, et al. Beyond seeing: Evaluating multimodal llms on tool-enabled image perception, transformation, and reasoning. *arXiv preprint arXiv:2510.12712*, 2025.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R Fung. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*, 2026a.

Yanna Jiang, Delong Li, Haiyu Deng, Baihe Ma, Xu Wang, Qin Wang, and Guangsheng Yu. Sok: Agentic skills–beyond tool use in llm agents. *arXiv preprint arXiv:2602.20867*, 2026b.

Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Chen Wei, Konstantinos Psounis, and Kaipeng Zhang. Tir-bench: A comprehensive benchmark for agentic thinking-with-images reasoning. In *European Conference on Computer Vision*, pp. 464–481. Springer, 2026a.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026b.

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. Skillc: Learning autonomous skill internalization in llm agents via contrastive credit assignment. *arXiv preprint arXiv:2605.27899*, 2026.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026a.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, et al. Skillrevise: Improving llm-authored agent skills via trace-conditioned skill revision. *arXiv preprint arXiv:2606.01139*, 2026b.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Krista Opsahl-Ong, Arnav Singhvi, Jasmine Collins, Ivan Zhou, Cindy Wang, Ashutosh Baheti, Owen Oertell, Jacob Portes, Sam Havens, Erich Elsen, et al. Officeqa pro: An enterprise benchmark for end-to-end grounded reasoning. *arXiv preprint arXiv:2603.08655*, 2026.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, et al. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*, 2026a.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026b.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Qiming Shi, Yibo Dou, Jiawen Zhu, Yulong Tao, Linbo Jin, Zhaolu Kang, Yunfan Zhou, and Di Weng. Skill-kd: Contrastive skill distillation for llm agents. *arXiv preprint arXiv:2607.28048*, 2026a.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

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

Zhaochen Su, Jincheng Gao, Hangyu Guo, Zhenhua Liu, Lueyang Zhang, Xinyu Geng, Shijue Huang, Peng Xia, Guanyu Jiang, Cheng Wang, et al. Agentvista: Evaluating multimodal agents in ultra-challenging realistic visual scenarios. *arXiv preprint arXiv:2602.23166*, 2026.

Xijia Tao, Teng Yihua, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. In *International Conference on Learning Representations*, volume 2026, pp. 70385–70422, 2026.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, et al. Skillx: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*, 2026a.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*, 2026b.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. Skillopt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026.

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

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

Zhuoyun Yu, Xin Xie, Wuguannan Yao, Chenxi Wang, Lei Liang, Xiang Qi, and Shumin Deng. Skilladaptor: Self-adapting skills for llm agents from trajectories. *arXiv preprint arXiv:2606.01311*, 2026.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026a.

Xiaoying Zhang, Zichen Liu, Yipeng Zhang, Xia Hu, and Wenqi Shao. Retroagent: From solving to evolving via retrospective dual intrinsic feedback. *arXiv preprint arXiv:2603.08561*, 2026b.

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

A PROMPT DESIGN

We list the prompt templates in Appendix. The templates below cover ALFWorld.

Teacher Hint for ALFWorld

```
You are a teacher reviewing a student agent's failed attempt at a household task.
    You can see the task goal and the student's full failed trajectory, but you
    must NOT spell out the exact sequence of actions that solves it.

Task goal: {goal}
Student's trajectory (actions + observations, ended WITHOUT completing the task):
{trajectory}

Generate a brief hint (1—2 sentences) that:
— Points the student toward what went wrong (e.g. "you kept revisiting locations
    you already searched")
— May name the category of mistake

Output the hint only, nothing else.
```

Policy Reflection for ALFWorld

```
Your attempt at household task FAILED (the task goal was not satisfied within the
    step budget).

Task goal: {goal}
Your trajectory (actions + observations):
{trajectory}
Hint from reviewer: {hint}

SECTION 1 ERROR ANALYSIS (execute in order):

STEP 1 — Search discipline:
  Did you revisit receptacles/surfaces you had already observed to lack the target?
  Did you restart a search class from the beginning instead of continuing at the
    next unsearched instance?

STEP 2 — Object & state handling:
  Did you pick up the exact requested object type (not a distractor)?
  If the task needed clean/heat/cool/use, did you perform the state change at the
    right appliance BEFORE delivery?
  Did you waste steps opening/toggling/examining appliances when a direct command
    was admissible?

STEP 3 — Decision & termination:
  Did you go straight to the remembered destination once holding the object?
  Did you stop prematurely, or loop the same action more than twice?

SECTION 2 GENERAL RULE:

Based on this failure, extract ONE general rule (not specific to this task
    instance, applicable to all similar ALFWorld tasks).

Format:
— Error pattern: One sentence describing the common pattern of this type of failure
— Applicable scenario: When is this failure most likely to occur
— Prevention method: Specific, actionable steps to avoid it
— Recommended procedure: The concrete action discipline to follow (search order,
    when to transform, when to deliver)

SECTION 3 NEXT ROUND CORRECTION:
1. What was the single most likely point of failure? (1 sentence)
2. What concrete change in your action policy should you make next round? (1—2
    sentences)
```

Reflection-Skill Consolidation for ALFWorld

```
You are a skill library architect for an EMBODIED TEXT AGENT (household tasks:
    pick_and_place, pick_two_obj_and_place, look_at_obj_in_light,
    pick_heat_then_place_in_recep, pick_cool_then_place_in_recep,
    pick_clean_then_place_in_recep). The agent chooses from an admissible action
    list at each step and outputs <think>...</think><action>...</action>.

Below are {N} lessons extracted from the agent's reflection logs. Each case is a
    game where the agent's first attempt FAILED (env.won=False) and a later
    attempt WITH reflection succeeded (env.won=True). Each case has:
  — task_type: ALFWorld task family
  — goal: the household task goal
  — trigger: a directional hint from the teacher (does NOT enumerate the solution)
  — Failure_pattern: how the agent failed on first try (trajectory tail)
  — How_to_repair: the GENERAL RULE the student LLM extracted in reflection
  — Verification: observable environment feedback or state changes used to verify
    that the error has been corrected
  — boundary: conditions under which the repair rule does not apply or requires
    adjustment

Your task: read all {N} lessons, group them into cohesive skill categories (each
    addressing one family of embodied failure modes specific to ALFWorld ), and
    produce a complete SKILL.md per category.
```

Test-Time Reflection for ALFWorld

```
You are an expert assistant for an embodied text agent on ALFWorld. Review the
    action trace of a PREVIOUS attempt at this task and patch the methodology to
    avoid common failure modes. The previous attempt's outcome is hidden from you
    on purpose.

### Your Goals:
1. Read Stage—1 Trace: below is the previous attempt's action/observation
    sequence. Identify risky or wasteful steps.
2. Apply Reflect Skills: check whether the trace commits any documented ALFWorld
    mistake in the reflect skills.
3. Produce a Reflect—Aware Guide: a focused procedure for REDOING this task,
    highlighting the pitfalls the prior attempt may have fallen into and the
    concrete corrective actions. Encourage systematic receptacle sweeping and
    verifying preconditions before placement.
4. Stay Lean: ~400 words max, markdown starting with `#`.

### Input:
<stage1_trace>
{stage1_summary}
</stage1_trace>

<reflect_skills>
{reflect_skills}
</reflect_skills>

<task>
{task}
</task>
```

B CASE STUDY

We present qualitative examples of how learned skills are used at inference time. The policy first attempts each task with the success skill, then uses the reflection skill to review its trajectory and generate a guide for a retry. We show some cases.

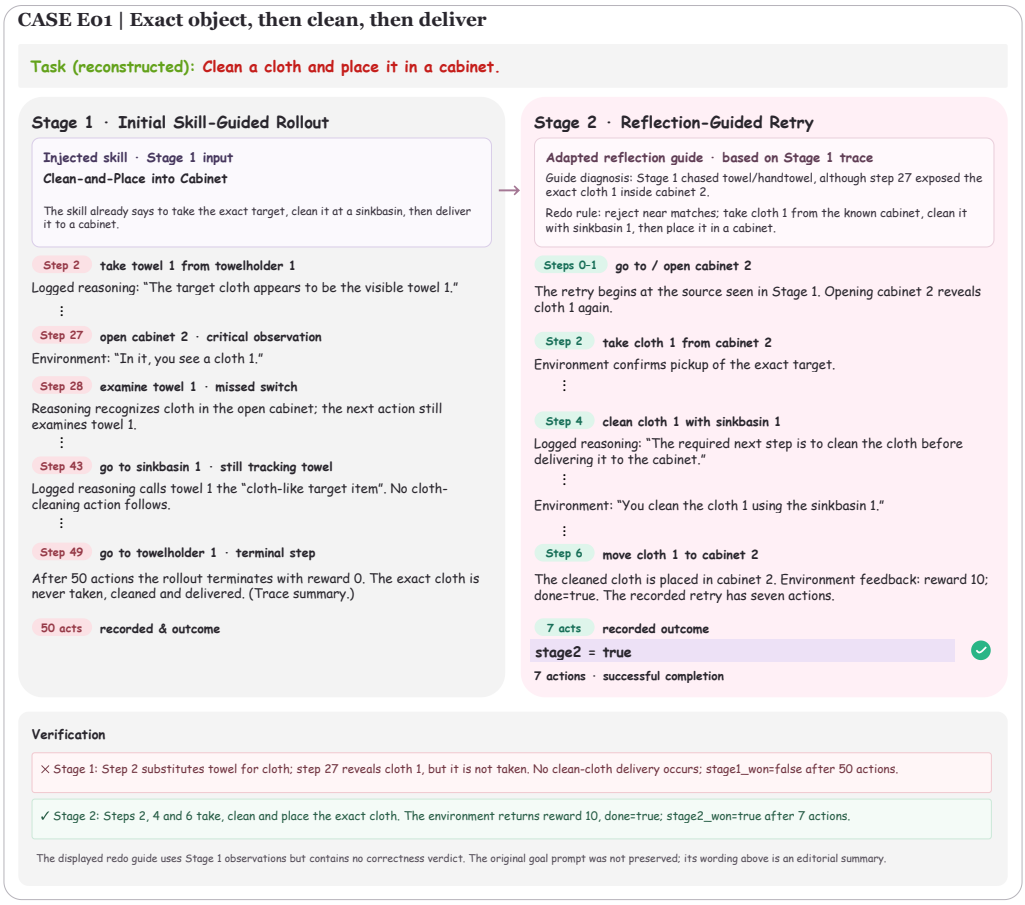


Figure 4: The initial attempt treats a towel as the requested cloth and fails to switch targets after observing cloth 1 in cabinet 2. An adapted reflection guide directs the retry to this previously observed object. The retry takes, cleans, and places the cloth in a cabinet, and the environment reports successful completion after seven actions. We show selected actions and observations; the task description is an editorial summary.

CASE X01 | Reopening a mountain-pass hypothesis

Question: What is the location?



Phase 1 · Initial Skill-Guided Rollout

Injected skill · Stage 1 input

Landmark and Signage Triangulation for Geolocation

Skill procedure: crop and read signs; search exact text; compare billboard, road infrastructure and mountain skyline before committing.

Turn 0 code_interpreter · image crops

Cropped the billboard, road signs, mountains and pylon for inspection.

Turn 1 web_search · exact billboard and phone

"暖盾" 姜茶 173-9966 → no results found.

Turn 6 web_search · candidate leap

"暖盾姜茶" 广告牌 金沙江大拐弯 梅里雪山

Turn 8 web_search · Baima hypothesis

"暖盾姜茶" 广告牌 白马雪山 → unrelated result.

Turn 10 recorded final explanation · excerpt

"The sharp, snow-capped peaks in the background belong to the Baima Snow Mountain range." The reply also attributes the pylon and road layout to Baima/G214.

Turn 10 final answer

Baima Snow Mountain Pass, G214, Yunnan

Phase 2 · Reflection-Guided Retry

Adapted reflection guide · based on Stage 1 trace

Guide diagnosis: Stage 1 followed one billboard cue and searched Jinshajiang, Meili and Baima in turns 6–8 without explicit image-to-place evidence.

Guide instruction: "Do not commit to a location until at least three independent visual markers align" — billboard, infrastructure and topography.

Turn 0 image_search · reverse search

Reverse search of the original frame; the returned hits did not identify the location.

Turn 1 code_interpreter · detail crops

Cropped billboard and road signs for another inspection.

Turn 10 image_search · query and returned text

"暖盾" 姜茶 广告牌 垭口

Returned result text names "川藏318折多山垭口".

Turn 12-15 image_search · compare alternatives

Turns 12, 14 and 15 searched Zheduo, Hongla and Baima separately. The saved trace records candidate comparisons, not an independent point-by-point image registration.

Turn 16 final answer

Zheduo Mountain Pass, Kangding, Sichuan

Recorded accuracy: 1.0

Verification

× Stage 1: The billboard was identified, but searches added Baima/Meili candidates without a confirming match. The final Baima/G214 answer differs from Zheduo (accuracy 0.0).

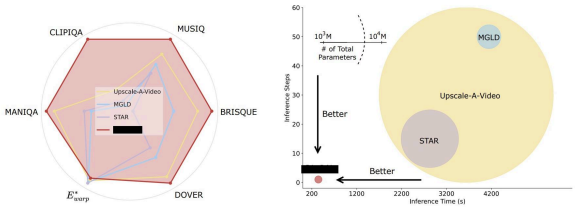
✓ Stage 2: A new search result names Zheduo/G318; later queries compare alternatives. The final Zheduo answer matches the recorded reference (accuracy 1.0).

Information boundary: Stage 2 received the adapted guide, not Stage 1's accuracy or ground-truth label. The trace does not independently prove every visual anchor matched.

Figure 5: The initial attempt commits to Baima Snow Mountain Pass despite inconclusive search results. An adapted reflection guide identifies unsupported assumptions in the initial trajectory and prompts further verification. The retry retrieves a result naming Zheduo Mountain Pass, compares alternative candidates, and returns an answer matching the reference. We show selected trajectory excerpts; correctness labels are post-run evaluation annotations and are not provided to the policy.

CASE X02 | Recovering the exact training-paradigm name

Question: What is the name of the paradigm proposed for training a diffusion model?



(a) Quality comparison (b) Efficiency Comparison

Phase 1 · Initial Skill-Guided Rollout

Injected success skill · Stage 1 input

Skill procedure: use figure-label anchors to find the source paper, then extract the exact training-paradigm name from the paper text.

Turn 0 **web_search · query**

"Upscale-A-Video" "MGLD" "STAR" diffusion model paper

Turn 1 **image_search (reverse) · returned title**

One-Step Diffusion for Detail-Rich and Temporally Consistent Video Super-Resolution

Turn 2 **visit · goal**

Identify the proposed paradigm name in the DOVE paper.

Turn 4 **web_search · query**

"DOVE" "latent-pixel training strategy" paradigm diffusion

Turn 5 **recorded final explanation · verbatim excerpt**

To effectively train this one-step diffusion model, the authors propose a specific training paradigm called the **latent-pixel training strategy** (or **latent-pixel training paradigm**).

Turn 5 **final answer**

latent-pixel training strategy

Phase 2 · Reflection-Guided Retry

Adapted reflection guide · based on Stage 1 trace

Guide diagnosis (hypothesis): "Stage-1 searched for a 'DOVE' paper. This was likely a misreading of the 'DOVER' metric label."

Guide instruction: "The question specifically asks for the name of the paradigm proposed for training, not just the name of the model or the paper title."

Turn 0 **web_search · query**

"Upscale-A-Video" "MGLD" "STAR" video super-resolution diffusion model "CLIPIQA" "DOVER"

Turn 1 **visit · returned error**

The Note VfT84F6YfH was not found

Turn 2 **web_search · query and returned snippet**

"One-Step Diffusion for Detail-Rich and Temporally Consistent Video Super-Resolution" training paradigm

To achieve this, we propose a Dual LoRA Learning (DLoRAL) paradigm

Turn 3 **recorded final explanation · verbatim excerpt**

The name of the paradigm proposed for training the diffusion model in this work is the **Dual LoRA Learning (DLoRAL)** paradigm.

Turn 3 **final answer**

Dual LoRA Learning (DLoRAL)

Recorded accuracy: 1.0

Verification

× Stage 1: Reverse search found the source title, but later searches pursued DOVE. "latent-pixel training strategy" did not match the reference answer (accuracy 0.0).

✓ Stage 2: An exact-title + "training paradigm" search returned the explicit DLoRAL snippet. "Dual LoRA Learning (DLoRAL)" matched the reference answer (accuracy 1.0).

Information boundary: These verdicts are post-run evaluator annotations. Stage 2 received the reflection guide, but not Stage 1's accuracy or ground-truth label.

Figure 6: The initial attempt retrieves the target paper's title but subsequently pursues a different paper and returns an incorrect training-strategy name. Using an adapted reflection guide, the policy retries the task, searches with the retrieved title, and obtains the correct paradigm name, Dual LoRA Learning (DLoRAL). We show selected trajectory excerpts; correctness labels are post-run evaluation annotations and are not provided to the policy.