CASU: CASE-TO-CLASS SKILL SYNTHESIS WITH ADAPTIVE SKILL UTILIZATION FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM) agents increasingly solve complex tasks through interaction, producing trajectories that could benefit future related tasks by preserving successful strategies and failure lessons. Yet this experience often remains underused: raw trajectories are too noisy and instance-specific to become transferable skills, and even when such skills are extracted, it remains unclear how to use them effectively, since some skills are better internalized into model parameters while others are better retained as external guidance. To address these challenges, we introduce **CASU**, a three-stage framework for hierarchical skill synthesis and adaptive skill utilization that uses only the agent's own experience, without stronger external supervision. Specifically, CASU distills multiple successful and failed trajectories of each task instance into *Case Skills*, which capture reusable strategies while filtering instance-specific details. It then synthesizes *Class Skills* from *Case Skills* collected across tasks in the same category, and optimizes the *Class Skill* generator with paired environment feedback on held-out tasks. It further proposes Skill Absorption Efficiency (SAE) to estimate the internalization suitability of each learned skill by measuring the consistency of its induced parameter updates across disjoint task subsets. High-SAE skills are internalized through distillation on skill-guided responses, while remaining useful skills are retained for external invocation. Across ALFWorld and WebShop, CASU improves GRPO-trained Qwen2.5-3B and Qwen2.5-7B agents, delivering up to 10.95 points of improvement in ALFWorld average success and 7.0 points in WebShop accuracy, surpassing strong RL, skill-augmented RL, and self-distillation baselines.

1 INTRODUCTION

Large language models (LLMs) have entered an agentic era. Frontier models such as GPT-6 Astra (OpenAI, 2026), Claude Fable 5.1 (Anthropic, 2026), and Qwen3.8-Max (Qwen Team, 2026), together with agent harnesses such as Codex (Bonamy & Choi, 2026), Claude Code (Anthropic, 2025), and DeepSeek Harness (DeepSeek-AI, 2026), exemplify this transition. As agents tackle recurring tasks, a central challenge is to retain useful experience and reuse it across interactions. *Agent skills* provide a practical way to organize this experience into reusable procedural knowledge. Introduced by Anthropic, *Agent Skills* package task-specific instructions, scripts, and supporting resources that agents can load when needed (Zhang et al., 2025). This perspective makes skills a natural unit for capturing effective procedures and transferring them across related tasks, motivating the study of how agents can acquire skills from their own experience.

As agents are deployed across diverse domains, equipping them with reliable skills becomes increasingly important. However, manually authoring and maintaining skills requires substantial expertise and effort (Ni et al., 2026), while retrieving and loading lengthy skill descriptions consumes context and adds inference-time token overhead (Lu et al., 2026b; Ma et al., 2026). To reduce authoring costs, recent methods synthesize skills from successful and failed trajectories and consolidate lessons across tasks (Ni et al., 2026; Wang et al., 2026a). Such synthesis must nevertheless process verbose, noisy interaction histories within limited context budgets and distinguish transferable procedures from details specific to individual episodes (Xia et al., 2026; Ni et al., 2026). To reduce dependence on runtime skill prompts, another line of work internalizes skill-guided behavior into model parameters (Lu et al., 2026b; Lin et al., 2026; Wang et al., 2026b). However, many of these methods rely on powerful external models such as OpenAI o3 and Seed1.8 for skill generation for small models, increasing the

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.

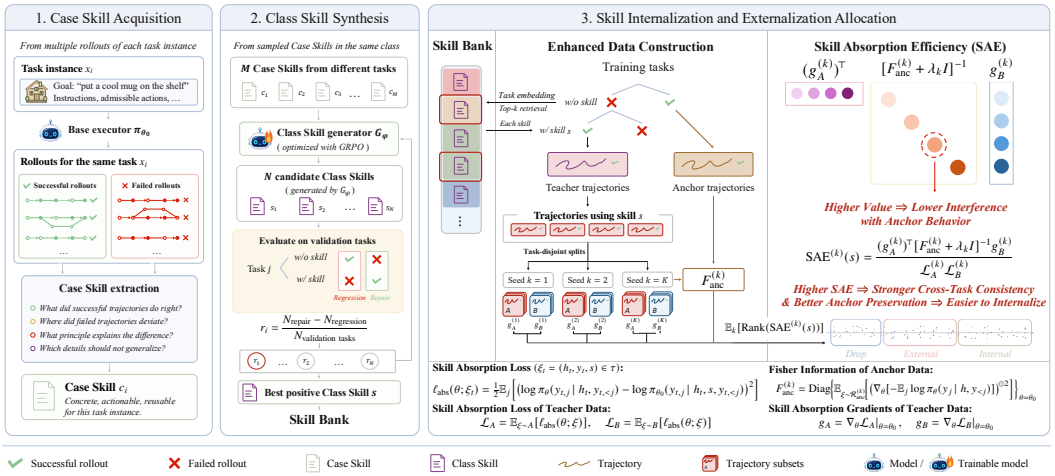

Figure 1: **The pipeline of CASU. (I) Case Skill Acquisition.** Repeated rollouts for each task are summarized into a Case Skill that captures strategies and mistakes. **(II) Class Skill Synthesis.** A generator consolidates Case Skills from the same category into Class Skills and is optimized with GRPO using feedback from separate validation tasks. **(III) Skill Internalization and Externalization Allocation.** Skill Absorption Efficiency (SAE) estimates internalization suitability by comparing gradients across disjoint task subsets under a Fisher metric from successful unassisted responses. SAE ranks guide allocation: high-ranked skills are selected for distillation, intermediate-ranked skills are selectively retained for external invocation, and particularly low ones are discarded.

cost of skill acquisition. Moreover, removing runtime skill access does not justify internalizing every skill. Internalized task-specific patterns can conflict with other procedures, causing performance regressions (Zhu et al., 2026; Lin et al., 2026). These limitations raise a central question: Can transferable skills be acquired from an agent's own experience without a stronger external teacher and then selectively internalized or retained as external guidance according to their internalization suitability?

To answer this question, we propose CASU (**C**ase-to-Class Skill Synthesis with **A**daptive **S**kill **U**tilization), a three-stage framework for hierarchical skill synthesis and adaptive skill utilization. As illustrated in Figure 1, CASU first summarizes repeated attempts at each task into a *Case Skill*, extracting effective strategies and mistakes to avoid. Since these local lessons do not necessarily generalize across instances, a *Class Skill* generator is trained to synthesize transferable procedures from *Case Skills* within the same task category, where Group Relative Policy Optimization (GRPO; Shao et al., 2024) is utilized. Each generated skill is evaluated on validation tasks from that category, with each task executed both with and without skill guidance. The reward favors skills that turn failed attempts into successful ones and penalizes those causing previously successful attempts to fail. However, effectiveness as external guidance does not mean suitability for internalization: learning a skill may require conflicting parameter updates across tasks or interfere with existing abilities. We therefore introduce *Skill Absorption Efficiency* (SAE), a measure of internalization suitability that compares skill-absorption gradients across two disjoint task subsets. Gradient agreement indicates that different tasks support compatible parameter updates. To account for potential interference, SAE uses a Fisher metric estimated from successful unassisted anchor responses to downweight parameter directions to which existing behavior is sensitive. We then internalize selected high-SAE-rank skills through offline distillation, training the agent to reproduce skill-guided responses without access to the skill text. Among the remaining skills, those with particularly low ranks are discarded, while those with intermediate ranks are selectively retained for external invocation.

We evaluate CASU using the Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct models on ALFWorld and WebShop. Initialized with a GRPO-trained model, CASU consistently improves over GRPO (+10.95/+9.28 on ALFWorld and +7.0/+2.4 on WebShop accuracy for 3B/7B), surpassing the compared reinforcement learning, skill-augmented reinforcement learning, and self-distillation baselines and achieving the best aggregate performance across both benchmarks and model scales.

2 Related Work

Skill synthesis and external memory. Automatic skill synthesis reduces manual authoring costs. Reflexion (Shinn et al., 2023) and ExpeL (Zhao et al., 2024) retain feedback and guidelines, while Voyager (Wang et al., 2024) builds executable skills. ReasoningBank (Ouyang et al., 2026), Trace2Skill (Ni et al., 2026), and SkillX (Wang et al., 2026a) organize trajectory experience into transferable knowledge. SkillGen (Ma et al., 2026) and Skill-R1 (Vishe et al., 2026) use execution feedback to improve skills; SkillRL (Xia et al., 2026), D2Skill (Tu et al., 2026), and Skill1 (Shi et al., 2026) couple skill evolution with policy learning. AutoSkill (Yang et al., 2026b) and Memento-Skills (Zhou et al., 2026) maintain external skills through continued interaction. However, skill synthesis often uses proprietary models such as o3, Seed1.8 or larger open models such as Qwen3.5-35B and Qwen3.5-122B, which can increase generation costs. CASU instead learns cross-task synthesis from compressed task-level lessons using copies of the executor and environment feedback, without a stronger external teacher.

Skill internalization and selective utilization. Skill internalization transfers guided behavior into parameters to reduce retrieval and context costs. Building on privileged-information distillation (Lopez-Paz et al., 2016), SKILL0 (Lu et al., 2026b) withdraws skill context, SKILLC (Lin et al., 2026) assigns learning credit using outcome differences, and Skill-SD (Wang et al., 2026b) distills skill-conditioned token distributions. SDAR (Lu et al., 2026a) and SIRI (He et al., 2026) selectively distill beneficial guidance. Skill0.5 (Zhu et al., 2026) combines general-skill internalization with task-specific guidance and adapts training to task difficulty. SLIM (Shen et al., 2026) retains or retires skills according to their marginal external contribution and expands its bank to address persistent failures. CASU instead selects internalization targets by cross-task update consistency and interference risk, assessing learning suitability beyond skill type or external utility.

3 Method

CASU learns reusable skills from interaction experience and determines how to incorporate them into subsequent task execution. As illustrated in Figure 1, the framework proceeds in three stages: *Case Skill Acquisition* compresses repeated attempts at individual tasks into local lessons; *Class Skill Synthesis* learns to combine these lessons into procedures that improve performance on other tasks in the same category; and *Skill Internalization and Externalization Allocation* selects which procedures to learn through parameter updates and which to retain as external guidance. Algorithm 1 in Appendix B.1 summarizes the complete procedure.

Problem setup. Let x_i be a task in category κ, and let π_{θ_0} denote the initial model. Given interaction history h_t, the model generates a response y_t and receives feedback from the environment. A trajectory τ records these interactions, with success indicator $Y(\tau) \in \{0, 1\}$. Using a skill s gives the policy $\pi_{\theta_0}(\cdot \mid h_t, s)$; without it, the model acts from the interaction history alone. Source tasks $\mathcal{D}_{\mathrm{src}}$ provide the tasks used for skill synthesis and internalization. Separate reward-validation tasks $\mathcal{D}_{\mathrm{rv}}$ measure the behavioral benefit of generated skills. All synthesis and teacher models are initialized from the same checkpoint, so skill acquisition requires no stronger external teacher. The final outputs are a trained model $\pi_{\theta^\star}$ and an external skill bank $\mathcal{B}^\star_{\mathrm{ext}}$.

3.1 Case Skill Acquisition

Raw trajectories contain useful decisions alongside repeated observations, unsuccessful actions, and details specific to the current task. We first organize this evidence at the task level. For each source task x_i, the frozen model collects R rollouts, and a frozen copy summarizes them into a *Case Skill*:

$$c_i = g_{\mathrm{case}}\big(x_i, \{(\tau_{i,j}, Y(\tau_{i,j}))\}_{j=1}^{R}\big)\,. \tag{1}$$

Guided by the four questions in Figure 1, the prompt distills each task's rollouts into a concise *Case Skill* that captures effective strategies, mistakes to avoid, and applicability conditions while removing instance-specific details. These summaries support subsequent cross-task *Class Skill* synthesis.

3.2 CLASS SKILL SYNTHESIS

Generalizing from individual cases requires identifying strategies shared across tasks. We therefore learn to synthesize *Class Skills* from multiple *Case Skills*, using their effect on separate tasks as supervision. For each synthesis input $u = \{c_{i_1}, \ldots, c_{i_M}\}$, the M cases come from distinct source tasks in the same category. A generator G_ϕ, initialized from π_{θ_0}, samples a group of N candidate Class Skills $s_1, \ldots, s_N$. Each candidate abstracts a shared procedure rather than concatenating cases.

Learning from behavioral gains. We evaluate each candidate s on same-category tasks from $\mathcal{D}_{\mathrm{rv}}$, running the frozen model both with and without that skill under matched execution conditions. Let $Y_i(s)$ and $Y_i(\varnothing)$ denote the two success indicators. Across N_κ paired validation executions, including evaluation repetitions, the reward is

$$U_{\mathrm{rv}}(s) = \frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} \big[Y_i(s) - Y_i(\varnothing)\big] = \frac{N_{\mathrm{repair}}(s) - N_{\mathrm{regression}}(s)}{N_\kappa}. \tag{2}$$

A repair occurs when skill guidance turns failure into success; a regression occurs when it turns success into failure. The reward thus measures whether the proposed procedure helps beyond the cases used to synthesize it, while accounting for harm to tasks the model could already solve.

We train G_ϕ with Group Relative Policy Optimization (GRPO; Shao et al., 2024), using rewards $r_j = U_{\mathrm{rv}}(s_j)$ normalized within each candidate group to obtain advantages. GRPO updates the generator on its skill tokens; the model that executes validation tasks remains frozen. For each group, we admit the highest-reward candidate to the category-indexed Class Skill Bank if its reward is positive and it is not a duplicate. We then select and freeze a bank checkpoint $\mathcal{B}^\star$ for Stage 3.

3.3 SKILL INTERNALIZATION AND EXTERNALIZATION ALLOCATION

Positive execution gains establish a skill's value as external guidance, but do not establish whether its behavior can be learned consistently across tasks. A skill may induce conflicting parameter updates on different instances, or require changes along directions important to existing abilities. We introduce *Skill Absorption Efficiency* (SAE) to guide this allocation.

Constructing learning and retention examples. For source tasks, we retrieve same-category skills from $\mathcal{B}^\star$ and collect paired executions with and without each candidate. Skill-guided trajectories provide teacher responses when they repair a failure or shorten an already successful execution. Successful unassisted trajectories provide *anchor responses*, representing behavior to retain. We group teacher responses by the skill that generated them and compute source utility $U_{\mathrm{src}}(s)$ using Equation 2 on the corresponding source pairs. Shorter successful executions supply learning examples but do not change this binary success-based utility. All internalization examples come from $\mathcal{D}_{\mathrm{src}}$; reward-validation trajectories are not used as distillation targets.

Measuring cross-task learning compatibility. For a stored teacher response $\xi = (h, y, s)$, we compare the probabilities assigned to its tokens by a skill-free student and the frozen, skill-guided teacher. The skill absorption loss is

$$\ell_{\mathrm{abs}}(\theta; \xi) = \frac{1}{2}\mathbb{E}_j\Big[\big(\log \pi_\theta(y_j \mid h, y_{<j}) - \log \pi_{\theta_0}(y_j \mid h, s, y_{<j})\big)^2\Big], \tag{3}$$

where $\mathbb{E}_j$ averages the response tokens. Both policies score the same stored response, with gradients computed only for the student. Minimizing this loss aligns student and teacher token probabilities.

For each seed k, we split supporting tasks into disjoint halves A and B, keeping each task's responses together. For responses $\mathcal{R}^{(k)}_{s,H}$ in half $H \in \{A, B\}$, we compute

$$\mathcal{L}^{(k)}_{s,H}(\theta) = \mathbb{E}_{\xi \sim \mathcal{R}^{(k)}_{s,H}}\big[\ell_{\mathrm{abs}}(\theta; \xi)\big], \qquad \mathbf{g}^{(k)}_{s,H} = \nabla_\theta \mathcal{L}^{(k)}_{s,H}(\theta)\Big|_{\theta=\theta_0}. \tag{4}$$

The expectation averages the stored responses in that half. The two gradients describe how the same initial model would change when learning the skill from different tasks. Agreement supports a shared learning direction; conflict indicates that the examples demand incompatible local updates. Below, $\mathcal{L}^{(k)}_{s,H}$ without an argument denotes its value at θ_0.

Accounting for interference with existing behavior. Gradient agreement alone does not account for the model's existing abilities. We therefore estimate a diagonal empirical Fisher matrix from successful unassisted responses. Let $\ell_{\mathrm{anc}}(\theta;\xi)$ be the mean negative log-likelihood of an anchor response. For each partition seed, define

$$\mathbf{F}_{\mathrm{anc}}^{(k)} = \mathrm{Diag}\Big\{\mathbb{E}_{\xi\sim\mathcal{R}_{\mathrm{anc}}^{(k)}}\Big[\big(\nabla_\theta \ell_{\mathrm{anc}}(\theta;\xi)|_{\theta=\theta_0}\big)^{\odot 2}\Big]\Big\}, \tag{5}$$

where $\odot 2$ denotes elementwise squaring. The damped inverse downweights directions important to anchor behavior. Applying this metric to the task-disjoint gradients gives

$$\mathrm{SAE}^{(k)}(s) = \frac{\big(\mathbf{g}_{s,A}^{(k)}\big)^\top \big[\mathbf{F}_{\mathrm{anc}}^{(k)} + \lambda_k \mathbf{I}\big]^{-1} \mathbf{g}_{s,B}^{(k)}}{\mathcal{L}_{s,A}^{(k)} \mathcal{L}_{s,B}^{(k)}}, \qquad \lambda_k > 0. \tag{6}$$

Appendix B.2 proves that, under the stated local smoothness conditions, positive SAE ensures that a loss-normalized local update from either task subset reduces the other subset's absorption loss for $0 < \eta < 2S/(\beta q_{\max})$. Here η is the step size of this local update, $S = \mathrm{SAE}^{(k)}(s)$, $\beta > 0$ bounds local curvature, and $q_{\max} = \max(q_A, q_B)$ is the larger squared norm of the two normalized gradients under the inverse anchor metric.

Allocating skills by consensus. We average normalized within-category SAE ranks over K task partitions to reduce sensitivity to individual splits:

$$\bar{r}_s = \mathbb{E}_k\Big[\mathrm{Rank}_\kappa\big(\mathrm{SAE}^{(k)}(s)\big)\Big]. \tag{7}$$

Among skills with positive utility and sufficient support, we internalize those with higher ranks and retain an intermediate-rank subset of the remainder as external candidates, discarding the rest. Appendix B.3 specifies eligibility criteria and thresholds.

Internalizing selected skills. Teacher responses associated with the selected skills form the internalization data. We mix these responses with unassisted anchors and train the student offline using the token-level skill distillation objective (SDL) of Skill-SD (Wang et al., 2026b):

$$\begin{aligned} \mathcal{L}_{\mathrm{SDL}}(\theta) &= \mathbb{E}_{\mathrm{tokens}}\big[\rho_\theta\big(e^{-\Delta_\theta} - 1 + \Delta_\theta\big)\big], \\ \Delta_\theta &= \log p_\theta - \log p_{\mathrm{T}}, \qquad \rho_\theta = \frac{p_\theta}{p_{\mathrm{old}}}. \end{aligned} \tag{8}$$

Here p_θ and p_{T} are token probabilities under the student and frozen teacher; p_{old} is the detached student probability rescored before each update. Responses stay fixed. Only the teacher receives skills on internalization examples; anchors use no skills. Appendix B.4 gives details.

Retaining complementary external guidance. After distillation, we reevaluate external candidates with $\pi_{\theta^\star}$ on $\mathcal{D}_{\mathrm{rv}}$ and retain those with sufficient net repairs, forming $\mathcal{B}_{\mathrm{ext}}^\star$. At inference, retrieval and reranking select at most one skill from this bank, with the option to select none.

4 Experiments

Our evaluation assesses three aspects of CASU: the effect of the complete pipeline on execution, the contribution of each synthesis stage, and the role of absorption evidence in skill selection. We study these aspects comprehensively through overall comparisons, ablations on synthesis and utilization, and analyses of training dynamics and representative cases.

4.1 Experimental Setup

Environments. We evaluate CASU on two interactive benchmarks. ALFWorld (Shridhar et al., 2020) is a text-based environment in which models navigate rooms and manipulate objects to complete household tasks, such as placing, cleaning, heating, and cooling objects. We report success rates across its six task categories and overall success (Avg). WebShop (Yao et al., 2022) simulates online shopping, requiring models to search for products, inspect their attributes, select options, and complete purchases that satisfy user requirements. We report mean environment reward (Score) and full-success rate (Acc), both as percentages. Our CASU evaluations use ALFWorld's `valid_seen` split and 500 shared WebShop task goals, with three attempts per task in both environments.

Table 1: **Performance on ALFWorld and WebShop.** All values are percentages. ALFWorld baselines are quoted from SDAR; CASU runs use three attempts per task on `valid_seen`. WebShop results are rerun on 500 fixed tasks with three attempts each. * indicates SDAR's static skills at inference. **Best** and second-best values are highlighted within each model block, including ties.

	ALFWorld							WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg	Score	Acc
Qwen2.5-3B-Instruct									
Vanilla	44.4	11.1	6.2	15.4	28.6	12.5	21.9	5.27	1.2
Skill-Prompt*	51.7	66.7	48.4	0	4.3	10	28.9	3.09	1
OPSD	48.8	41.7	16.7	0	15.8	16.7	28.1	2.82	1.2
GRPO	91.2	62.5	96.2	61.9	65	47.4	75	70.55	65.4
Skill-GRPO	88.9	71.4	58.8	70.6	40.7	29.2	60.2	75.71	66.8
Skill-GRPO*	94.3	57.1	**100**	66.7	73.1	57.1	80.5	80.31	70.6
GRPO+OPSD	**100**	**82.4**	85.7	75	70	60	81.2	75.11	56.2
Skill-SD	88.2	50	96.2	52.4	65	57.9	73.4	75.56	61.8
RLSD	87.9	75	90.9	75	73.1	68.4	79.7	82.28	66.8
SDAR	97.1	62.5	**100**	61.9	75	**84.2**	84.4	75.95	61
CASU	50.48	41.03	14.81	14.58	6.67	6.94	23.57	4.1	1.6
GRPO+CASU	96.19	74.36	92.59	**89.58**	**77.33**	76.39	**85.95**	**82.81**	**72.4**
Qwen2.5-7B-Instruct									
Vanilla	36.1	22.2	3.1	0	0	0	12.5	13.66	2.4
Skill-Prompt*	51.7	50	32.3	5.3	4.3	0	23.4	6.41	1.8
OPSD	50	60	22.7	21.4	17.6	9.5	32.8	9.87	1.6
GRPO	91.2	87.5	96.2	81	65	57.9	81.2	85.89	76.2
Skill-GRPO	88.5	66.7	65.2	61.1	57.7	73.1	69.5	84.18	71.4
Skill-GRPO*	**100**	83.3	96.4	83.3	75	78.9	88.3	81.94	70.8
GRPO+OPSD	91.4	61.5	**100**	87.5	76.5	52.2	80.4	85.87	76.8
Skill-SD	93.9	**93.8**	90.9	**100**	69.2	68.4	85.1	76.6	69
RLSD	**100**	87.5	92.3	58.8	**80**	65.2	82	84.45	76.4
SDAR	94.7	75	**100**	86.7	68.2	78.9	85.9	83.72	74.2
CASU	74.3	59	54.3	52.1	26.7	16.7	48.1	19.41	8.8
GRPO+CASU	**100**	74.36	98.77	93.75	**80**	**84.72**	**90.48**	**87.68**	**78.6**

Baselines. We adopt the baseline configurations used in SDAR (Lu et al., 2026a) and additionally include SDAR itself for comparison. We organize the baselines into three groups. First, *prompt-based methods* include Vanilla, which directly uses the instruction-tuned model, and Skill-Prompt, which augments inference prompts with retrieved skills. Second, *reinforcement learning methods* include GRPO (Shao et al., 2024) and Skill-GRPO, which incorporates retrieved skills into training prompts. Third, *self-distillation and hybrid methods* include OPSD (Zhao et al., 2026), which learns from a self-teacher with privileged context; GRPO+OPSD, which combines GRPO with an auxiliary self-distillation objective; Skill-SD (Wang et al., 2026b), which performs skill-conditioned self-distillation; RLSD (Yang et al., 2026a), which uses self-distillation signals to reweight token-level RL updates; and SDAR, which applies token-level gating to auxiliary self-distillation while preserving the RL objective. For these baselines, * denotes the use of external skills at inference.

Implementation Details. We use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Qwen Team, 2025) as base models and independently train their GRPO checkpoints for 150 steps. Applying CASU to raw instruct and our GRPO-trained checkpoints yields CASU and GRPO+CASU, respectively. Trainable baselines also use 150 steps, following the configurations of SDAR (Lu et al., 2026a). For CASU, skill retrieval combines embedding-based Top-5 selection with model-based reranking. Training hyperparameters, retrieval settings, and evaluation protocols are detailed in Appendix C.

4.2 Overall Performance

Best overall performance across both benchmarks. As shown in Table 1, GRPO+CASU achieves the highest ALFWorld average success and WebShop Score and Acc at both model scales. On WebShop, Acc increases from 65.4 to 72.4 for 3B and from 76.2 to 78.6 for 7B, gains of 7.0 and 2.4 over GRPO. Score improves by 12.26 and 1.79, respectively. On ALFWorld, success reaches 85.95

and 90.48, exceeding GRPO results by 10.95 and 9.28. Overall, these results demonstrate that CASU can effectively synthesize skills without external teacher and utilize them for stronger task execution.

CASU improves instruction models and complements RL. Without GRPO initialization of the execution model, CASU outperforms OPSD on both benchmarks at 7B: ALFWorld success reaches 48.10 compared with 32.80, while WebShop Score/Acc reach 19.41/8.8 compared with 9.87/1.6. At 3B, CASU also improves WebShop Score/Acc over OPSD, achieving 4.10/1.6 compared with 2.82/1.2. These results show that CASU is effective even without RL initialization, and that its gains are further amplified when applied on top of a GRPO-trained policy, which provides a stronger starting model for collecting useful experience.

4.3 Ablation Study

4.3.1 From Skill Synthesis to Selective Utilization

To assess stage contributions and adaptive skill utilization, we remove individual stages and compare allocation strategies. All ablations use a GRPO-initialized ALFWorld-7B model.

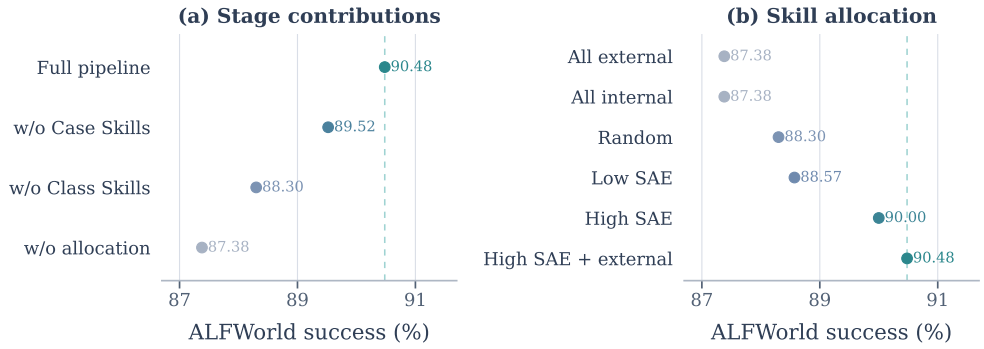

Figure 2: Ablation results on ALFWorld-7B, evaluating the effects of (a) the three pipeline stages and (b) skill selection and utilization strategies.

All three stages contribute to performance. As shown in Figure 2(a), replacing Case Skills with raw trajectories gives 89.52, while bypassing Class Skill synthesis and passing the Case Skill bank directly to allocation gives 88.30. Both are below the selected complete system, which scores 90.48. Using the full Class Skill bank externally without the allocation and distillation stage gives 87.38. Removing Case Skill synthesis, Class Skill synthesis, or selective utilization reduces performance, demonstrating the complementary value of all three stages. Appendix C.2 details the ablation settings.

SAE-guided allocation improves performance. As shown in Figure 2(b), high-SAE selection achieves 90.00, outperforming random and low-SAE selection, which score 88.30 and 88.57, respectively, with the same skill counts per category. Internalizing the full bank or using it entirely externally achieves 87.38 in both settings. Adding calibrated external skills to the high-SAE student further improves success to 90.48. These results show that SAE-based selection is critical: it outperforms random and low-SAE selection, while neither full internalization nor full externalization is sufficient, and combining high-SAE internalization with calibrated external skills yields the best performance.

4.3.2 Balancing Skill Coverage and Behavioral Retention

To examine how distillation balances skill coverage and behavioral retention, we vary two factors: the within-category SAE threshold q, which controls how many skills are internalized, and the anchor fraction, which controls the share of supervision devoted to preserving existing behavior.

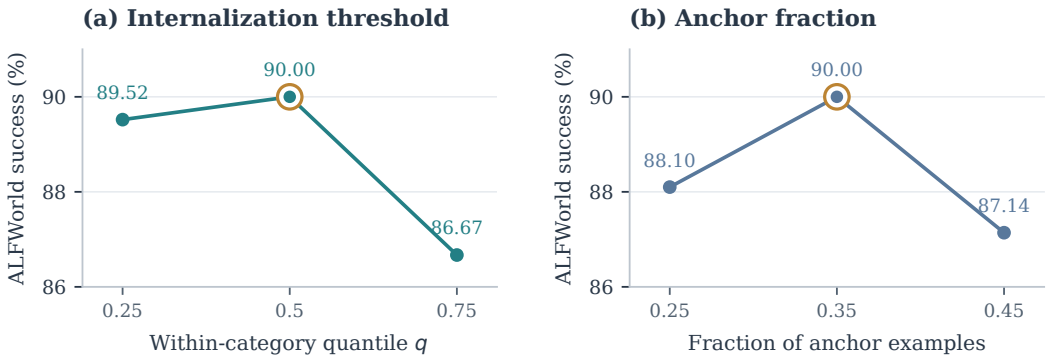

Figure 3: Sensitivity results on ALFWorld-7B, varying (a) the internalization threshold and (b) the anchor fraction. Outlined markers indicate the defaults.

Moderate settings best balance coverage and retention. As shown in Figure 3, the default threshold $q = 0.50$ and anchor fraction of 0.35 achieve the highest success rate of 90.00 among the tested settings. Relaxing the threshold to 0.25 changes success only slightly to 89.52, whereas raising it to 0.75 reduces success to 86.67, indicating that overly strict selection sacrifices useful skill coverage. Anchor fractions of 0.25 and 0.45 score 88.10 and 87.14, respectively, supporting a balance between learning new skills and preserving existing behavior.

4.4 Analysis

4.4.1 Training Dynamics

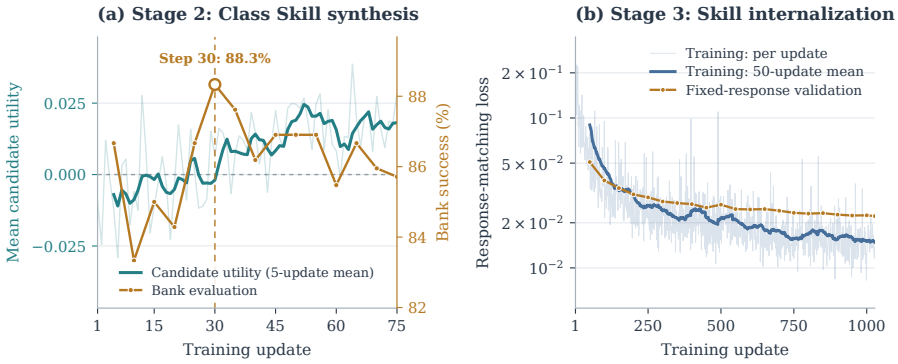

Figure 4: Training dynamics on ALFWorld-3B: (a) skill utility and bank success during synthesis; (b) training and validation response-matching loss during internalization. The dashed line marks the bank selected by validation success.

To examine skill learning, we analyze the dynamics of skill synthesis and internalization. As shown in Figure 4, during synthesis, mean skill utility increases from -0.0079 over the first ten updates to 0.0178 over the last ten, indicating that the generator progressively produces more useful skills. Bank success peaks at 88.33% at update 30, and we select this checkpoint for internalization. During internalization, both training and held-out response losses decrease steadily: mean training loss drops from 0.0898 over the first 50 updates to 0.0148 over the last 50, while held-out loss falls from 0.0508 to 0.0222. These trends show that synthesis improves skill utility and internalization effectively aligns the student with the teacher. Appendix C.3 provides additional curves and plotting details.

4.4.2 CASE STUDY

To examine how skills guide execution, we compare paired trajectories with and without skill guidance. As illustrated in Figure 5, the heating procedure selected for internalization prevents the model from cooling a heated potato, while the external location cue redirects search to the sofa to retrieve both pillows. These examples illustrate that the two allocation modes serve complementary roles: the internalized procedure corrects the action sequence by preventing an erroneous state change, whereas the externally retained cue supplies task-specific location knowledge that the unassisted policy lacks. This supports CASU's allocation principle that skills suitable for stable parameter learning are internalized, while skills providing contextual hints remain external, allowing both to repair different execution errors. Case context and allocation details are provided in Appendix D.1.

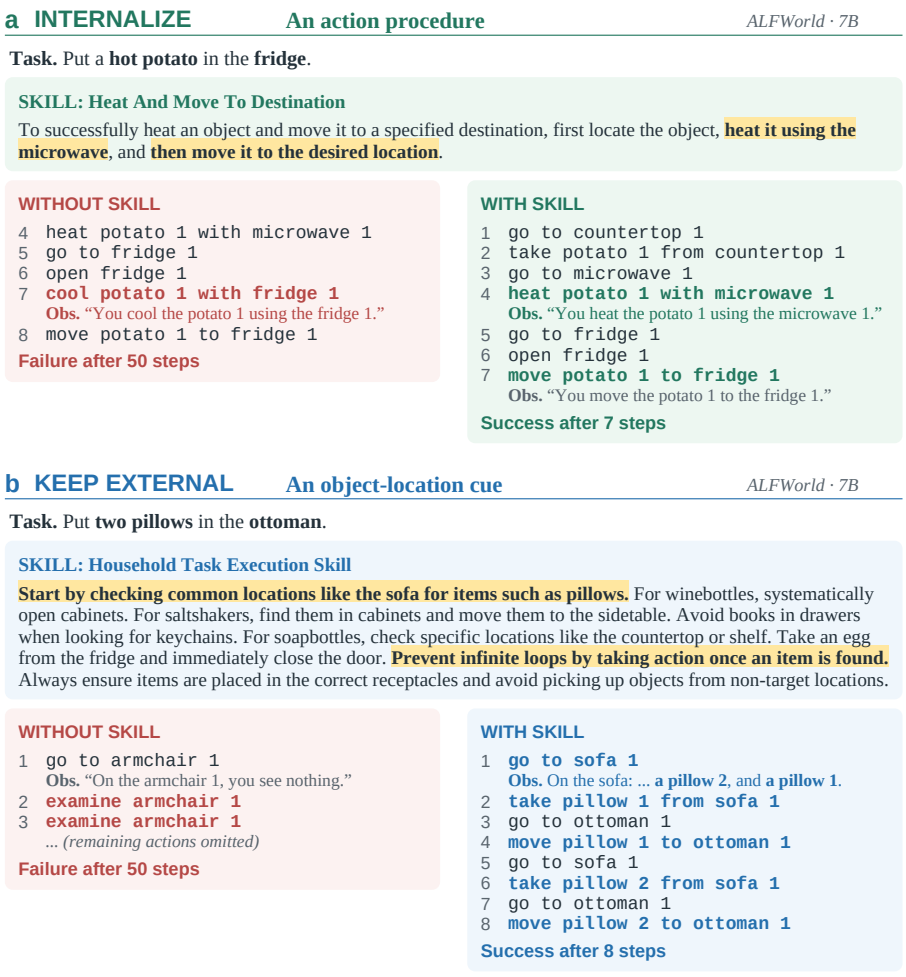

Figure 5: ALFWorld-7B cases showing (a) a heating procedure selected for internalization and (b) a location cue retained externally. Complete skills and paired traces show how guidance repairs failures.

5 CONCLUSION

We presented CASU, a three-stage framework for hierarchical skill synthesis and selective internalization that learns from an agent's own experience without a stronger external teacher. CASU distills task rollouts into Case Skills, synthesizes Class Skills with paired feedback, and uses Skill Absorption Efficiency (SAE) to decide which skills to internalize, retain externally. Experiments on ALFWorld and WebShop show that CASU improves GRPO-trained Qwen2.5-3B and Qwen2.5-7B agents by up to 10.95 and 7.0 points, respectively, surpassing strong RL, skill-augmented RL, and self-distillation baselines.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology details. Additionally, we used generative AI tools for modifing scientific figures, editing software code for debugging and summarizing existing literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. Figures and literatures are also checked manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude 3.7 Sonnet and Claude Code, February 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet.

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1, September 2026. URL https://www.anthropic.com/claude-fable-and-mythos-5-1.

Nicolas Bonamy and Derrick Choi. Codex as a Platform: Build on the Open Agent Harness. OpenAI Developer Blog, August 2026. URL https://developers.openai.com/blog/codex-as-a-platform.

DeepSeek-AI. DeepSeek Harness: Everything is a plugin, 2026. URL https://github.com/deepseek-ai/deepseek-harness.

Zhongyu He, Yuanfan Li, Fei Huang, Tianyu Chen, Siyuan Chen, Xingyang Li, Meng Hsuan Yu, Xiangrong Liu, Leyi Wei, Lu Pan, Ke Zeng, and Xunliang Cai. SIRI: Self-Internalizing Reinforcement Learning with Intrinsic Skills for LLM Agent Training. *arXiv preprint arXiv:2606.02355*, 2026. URL https://arxiv.org/abs/2606.02355.

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. URL https://arxiv.org/abs/2605.27899.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying Distillation and Privileged Information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026a. URL https://arxiv.org/abs/2605.15155.

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*, 2026b. URL https://arxiv.org/abs/2604.02268.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. SkillGen: Verified Inference-Time Agent Skill Synthesis. *arXiv preprint arXiv:2605.10999*, 2026. URL https://arxiv.org/abs/2605.10999.

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. URL https://arxiv.org/abs/2603.25158.

OpenAI. GPT-6 Astra. Official model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-6-astra.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25140.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Junhao Shen, Teng Zhang, Xiaoyan Zhao, and Hong Cheng. Dynamic Skill Lifecycle Management for Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.10923*, 2026. URL https://arxiv.org/abs/2605.10923.

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*, 2026. URL https://arxiv.org/abs/2605.06130.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36, 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. URL https://arxiv.org/abs/2010.03768.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yaocheng Zhang, Xiangyuan Lan, Linjing Li, Dong Li, and Dongbin Zhao. Dynamic Dual-Granularity Skill Bank for Agentic RL. *arXiv preprint arXiv:2603.28716*, 2026. URL https://arxiv.org/abs/2603.28716.

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A. Rossi, Jingbo Shang, Julian McAuley, and Junda Wu. Skill-R1: Agent Skill Evolution via Reinforcement Learning. *arXiv preprint arXiv:2605.09359*, 2026. URL https://arxiv.org/abs/2605.09359.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, and Shumin Deng. SkillX: Automatically Constructing Skill Knowledge Bases for Agents. *arXiv preprint arXiv:2604.04804*, 2026a. URL https://arxiv.org/abs/2604.04804.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-Conditioned Self-Distillation for Multi-turn LLM Agents. *arXiv preprint arXiv:2604.10674*, 2026b. URL https://arxiv.org/abs/2604.10674.

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. SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning. *arXiv preprint arXiv:2602.08234*, 2026. URL https://arxiv.org/abs/2602.08234.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-Distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026a. URL https://arxiv.org/abs/2604.03128.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. AutoSkill: Experience-Driven Lifelong Learning via Skill Self-Evolution. *arXiv preprint arXiv:2603.01145*, 2026b. URL https://arxiv.org/abs/2603.01145.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *arXiv preprint arXiv:2207.01206*, 2022. URL https://arxiv.org/abs/2207.01206.

Barry Zhang, Keith Lazuka, and Mahesh Murag. Equipping Agents for the Real World with Agent Skills. Anthropic Engineering, October 2025. URL https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills.

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*, 2024. URL https://arxiv.org/abs/2308.10144.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

Huichi Zhou, Siyuan Guo, Anjie Liu, Zhongwei Yu, Ziqin Gong, Bowen Zhao, Zhixun Chen, Menglong Zhang, Yihang Chen, Jinsong Li, Runyu Yang, Qiangbin Liu, Xinlei Yu, Jianmin Zhou, Na Wang, Chunyang Sun, and Jun Wang. Memento-Skills: Let Agents Design Agents. *arXiv preprint arXiv:2603.18743*, 2026. URL https://arxiv.org/abs/2603.18743.

Jiapeng Zhu, Jianxiang Yu, Yibo Zhao, Chengcheng Han, Qi Gu, Xunliang Cai, Xiang Li, and Weining Qian. Skill0.5: Joint Skill Internalization and Utilization for Out-of-Distribution Generalization in Agentic Reinforcement Learning. *arXiv preprint arXiv:2605.28424*, 2026. URL https://arxiv.org/abs/2605.28424.

A NOTATION

We distinguish losses from datasets and empirical utility from probability ratios. The sample-level absorption loss ℓ_{abs} is averaged to obtain $\mathcal{L}^{(k)}_{s,H}(\theta)$; its gradient is evaluated at θ_0. The same aggregate loss evaluated at θ_0 supplies the SAE denominator. All occurrences in the local-transfer derivation and the numerical implementation use this convention.

Symbol	Meaning
x_i, c_i, κ, s	Task, its Case Skill, task category, and Class Skill.
$\theta, \theta_0, \theta^\star; \phi$	Student, initial, trained, and skill-generator parameters.
$\mathcal{D}_{\mathrm{src}}, \mathcal{D}_{\mathrm{rv}}$	Source tasks and reward-validation tasks.
R, M, N, K	Attempts per task, Case Skills per synthesis prompt, candidate Class Skills per group, and SAE seeds.
$\xi = (h, y, s), T_\xi$	Response sample and its number of tokens; anchors omit skill context.
$\mathcal{R}^{(k)}_{s,H}$	Responses for skill s, task half $H \in \{A, B\}$, and seed k.
$\ell_{\mathrm{abs}}, \ell_{\mathrm{anc}}$	Per-response absorption loss and anchor negative log-likelihood.
$\mathcal{L}^{(k)}_{s,H}(\theta)$	Mean absorption loss over $\mathcal{R}^{(k)}_{s,H}$.
$\mathcal{L}^{(k)}_{s,H}, \mathbf{g}^{(k)}_{s,H}$	Initial loss value and its gradient at θ_0.
$\mathbf{F}^{(k)}_{\mathrm{anc}}, \mathbf{W}^{(k)}$	Diagonal anchor Fisher and its damped inverse.
$U_{\mathrm{rv}}(s), U_{\mathrm{src}}(s)$	Empirical skill utility on validation and retrieved source pairs.
$r^{(k)}_s, \bar{r}_s, m_\kappa$	Per-seed normalized rank, consensus rank, and eligible-category median.
$\mathcal{E}_\kappa, \mathcal{I}_\kappa, \mathcal{C}_{\mathrm{ext}}$	Eligible skills, internal skills, and external candidates.
$\mathcal{B}^\star, \mathcal{B}^\star_{\mathrm{ext}}$	Frozen Class Skill Bank and calibrated external bank.
$p_\theta, p_{\mathrm{T}}, p_{\mathrm{old}}$	Token probabilities: student, frozen teacher, pre-update student.
$\Delta_\theta, \rho_\theta$	Student–teacher log-probability difference and student probability ratio.
$\mathcal{L}_{\mathrm{SDL}}(\theta)$	Aggregate offline distillation objective.

B Additional Method Details

B.1 End-to-End Procedure

Algorithm 1 summarizes CASU. Source tasks provide Case Skills and all response-level data for SAE and distillation. Disjoint reward-validation tasks supply synthesis rewards and post-training external calibration. Bank selection uses $\mathcal{D}_{\mathrm{sel}}$; its relationship to reported evaluation splits is documented in Appendix C.1.

Algorithm 1 CASU: hierarchical skill synthesis and adaptive utilization

Input: Initial model π_{θ_0}; source tasks $\mathcal{D}_{\mathrm{src}}$; reward-validation tasks $\mathcal{D}_{\mathrm{rv}}$; bank-selection tasks $\mathcal{D}_{\mathrm{sel}}$
Output: Trained model $\pi_{\theta^\star}$ and calibrated external bank $\mathcal{B}^\star_{\mathrm{ext}}$

Freeze the executor and Case Skill synthesizer at θ_0; initialize G_ϕ from θ_0 and $\mathcal{B} \leftarrow \varnothing$.
Stage I: Case Skill Acquisition
for each source task $x_i \in \mathcal{D}_{\mathrm{src}}$ **do**
  Collect R trajectories with π_{θ_0} and record their outcomes.
  Synthesize Case Skill c_i using Equation 1.
end for
Stage II: Class Skill Synthesis
for each generator update **do**
  Sample M Case Skills from distinct tasks in one category.
  Sample N candidate Class Skills from G_ϕ for each synthesis input.
  Evaluate each candidate with paired skill/no-skill executions on same-category $\mathcal{D}_{\mathrm{rv}}$ tasks.
  Update ϕ with GRPO using $U_{\mathrm{rv}}(s)$ (Equation 2).
  Add each group's highest-reward candidate to $\mathcal{B}$ if its utility is positive and it is not a duplicate.
  Periodically evaluate bank snapshots on $\mathcal{D}_{\mathrm{sel}}$.
end for
Select and freeze the best evaluated bank snapshot as $\mathcal{B}^\star$.
Stage III: Skill Internalization and Externalization Allocation
Retrieve same-category candidates for $\mathcal{D}_{\mathrm{src}}$; collect exact-skill/no-skill pairs and compute $U_{\mathrm{src}}(s)$.
Keep assisted repair/efficiency trajectories as teacher data and successful unassisted trajectories as anchors.
for each partition seed $k = 1, \ldots, K$ **do**
  Estimate $\mathbf{F}^{(k)}_{\mathrm{anc}}$ from anchors (Equation 5).
  Split each supported skill's source tasks into disjoint halves; compute losses and gradients at θ_0 (Equation 4).
  Compute $\mathrm{SAE}^{(k)}(s)$ (Equation 6) and within-category ranks.
end for
Average ranks across seeds to obtain $\bar{r}_s$ (Equation 7).
Form eligible sets $\mathcal{E}_\kappa$ using validation/source utility, retrieval evidence, and task support.
Select skills above the eligible-category median for internalization: $\mathcal{I} = \bigcup_\kappa \mathcal{I}_\kappa$ (Equation 20).
Keep remaining eligible skills in the configured intermediate rank band as $\mathcal{C}_{\mathrm{ext}}$; discard the rest.
Mix teacher responses for $\mathcal{I}$ with anchors; split the source data into training and validation sets by task.
Initialize the student at θ_0; optimize offline SDL on fixed training responses to obtain $\theta^\star$ (Equation 8).
Reevaluate each external candidate on $\mathcal{D}_{\mathrm{rv}}$ with paired skill/no-skill executions of $\pi_{\theta^\star}$.
Retain candidates meeting the net-repair threshold as $\mathcal{B}^\star_{\mathrm{ext}}$ (Equation 22).
return $\pi_{\theta^\star}, \mathcal{B}^\star_{\mathrm{ext}}$.

B.2 Anchor Metric and Local Transfer

Response-level estimation. The absorption loss in Equation 3 takes the square of the log-probability difference for each token before averaging over the response. Equation 4 then averages these response-level losses. The teacher and student score identical stored responses, and all absorption gradients are evaluated at the same initial parameters θ_0. Task-level partitioning prevents responses from the same task from appearing in both halves.

For an anchor response $\xi = (h, y)$ of length T_ξ, define

$$\ell_{\mathrm{anc}}(\theta; \xi) = -\mathbb{E}_j\left[\log \pi_\theta(y_j \mid h, y_{<j})\right] = -\frac{1}{T_\xi}\sum_{j=1}^{T_\xi} \log \pi_\theta(y_j \mid h, y_{<j}). \tag{9}$$

Equation 5 squares the gradient of this mean response loss and then averages across anchor responses, giving a diagonal empirical second moment of response-level gradients. Each seed reestimates this

matrix and shares it across skills. Sampling, damping, and numerical stabilization are specified in Appendix C.

Setup. Fix a skill s and partition seed k, and suppress these indices below. For $H \in \{A, B\}$, let $\mathcal{L}_H(\theta) = \mathcal{L}^{(k)}_{s,H}(\theta)$ and assume the initial loss $L_H = \mathcal{L}_H(\theta_0)$ is positive. Define the normalized loss and its gradient at the initial model by

$$f_H(\theta) = \frac{\mathcal{L}_H(\theta)}{L_H}, \qquad \mathbf{u}_H = \nabla f_H(\theta_0) = \frac{\mathbf{g}_H}{L_H}, \qquad f_H(\theta_0) = 1. \tag{10}$$

Thus a change in f_H measures the absorption loss change relative to its initial value. Write $\mathbf{M} = \mathbf{F}_{\text{anc}} + \lambda \mathbf{I}$, $\mathbf{W} = \mathbf{M}^{-1}$, and $\|\mathbf{v}\|^2_{\mathbf{M}} = \mathbf{v}^\top \mathbf{M}\mathbf{v}$. In this notation, Equation 6 becomes

$$S = \text{SAE}^{(k)}(s) = \mathbf{u}_A^\top \mathbf{W} \mathbf{u}_B, \qquad q_H = \|\mathbf{u}_H\|^2_{\mathbf{W}}. \tag{11}$$

The following results analyze this score through local updates at θ_0.

Proposition B.1 (Local optimization under the anchor metric)**.** *For $\lambda > 0$ and $\eta > 0$, the unique minimizer of*

$$Q_H(\delta) = \mathbf{u}_H^\top \delta + \frac{1}{2\eta} \|\delta\|^2_{\mathbf{M}} \tag{12}$$

is $\delta_H = -\eta \mathbf{W} \mathbf{u}_H$. The linear term is the first-order relative absorption loss change, while the quadratic term penalizes parameter movement according to coordinatewise anchor sensitivity and damping.

Proof. Let $\mathbf{a}_\xi = \nabla_\theta \ell_{\text{anc}}(\theta_0; \xi)$, with p indexing parameter coordinates. By Equation 5, for any nonzero $\mathbf{v}$,

$$\mathbf{v}^\top \mathbf{M} \mathbf{v} = \mathbb{E}_\xi \left[\sum_p a^2_{\xi,p} v_p^2 \right] + \lambda \|\mathbf{v}\|_2^2 > 0. \tag{13}$$

Hence $\mathbf{M}$ is positive definite. Completing the square gives

$$Q_H(\delta) = \frac{1}{2\eta} \|\delta + \eta \mathbf{W} \mathbf{u}_H\|^2_{\mathbf{M}} - \frac{\eta}{2} \|\mathbf{u}_H\|^2_{\mathbf{W}}, \tag{14}$$

whose unique minimizer is δ_H. Furthermore, $\delta^\top \mathbf{F}_{\text{anc}} \delta = \mathbb{E}_\xi[\sum_p (a_{\xi,p} \delta_p)^2]$: each term is the squared first-order effect of changing one parameter coordinate on the anchor-response loss. Larger diagonal entries therefore assign a higher cost to movement in the corresponding coordinates. □

Proposition B.2 (Cross-task relative loss reduction)**.** *Suppose f_A and f_B are twice continuously differentiable in a neighborhood $\mathcal{U}$ of θ_0. For the local updates in Proposition B.1,*

$$\frac{\mathcal{L}_B(\theta_0 + \delta_A) - L_B}{L_B} = -\eta S + O(\eta^2), \tag{15}$$

and the same expansion holds after exchanging A and B. If, in addition, $\nabla^2 f_H(\theta) \preceq \beta \mathbf{M}$ for both $H \in \{A, B\}$ and all $\theta \in \mathcal{U}$, where $\beta > 0$, and both update segments remain in $\mathcal{U}$, then

$$\begin{aligned} f_B(\theta_0 + \delta_A) &\le 1 - \eta S + \frac{\beta \eta^2}{2} q_A, \\ f_A(\theta_0 + \delta_B) &\le 1 - \eta S + \frac{\beta \eta^2}{2} q_B. \end{aligned} \tag{16}$$

Consequently, when $S > 0$, both cross-task updates strictly reduce the respective losses for

$$0 < \eta < \frac{2S}{\beta \max(q_A, q_B)}. \tag{17}$$

Proof. Taylor expansion at θ_0 gives

$$f_B(\theta_0 + \delta_A) - 1 = \mathbf{u}_B^\top \delta_A + O(\|\delta_A\|_2^2) = -\eta \mathbf{u}_B^\top \mathbf{W} \mathbf{u}_A + O(\eta^2).$$

Symmetry of $\mathbf{W}$ implies $\mathbf{u}_B^\top \mathbf{W}\mathbf{u}_A = S$, proving Equation 15 and its counterpart. Under the curvature assumption, the integral form of the remainder yields

$$f_B(\theta_0 + \delta_A) = 1 + \mathbf{u}_B^\top \delta_A + \int_0^1 (1-t)\delta_A^\top \nabla^2 f_B(\theta_0 + t\delta_A)\delta_A \, dt$$
$$\leq 1 - \eta S + \frac{\beta}{2}\|\delta_A\|_{\mathbf{M}}^2 = 1 - \eta S + \frac{\beta\eta^2}{2} q_A. \tag{18}$$

Exchanging the two subsets gives the second bound. If $S > 0$, both q_A and q_B are positive, and Equation 17 makes each upper bound strictly less than one. Since $L_A, L_B > 0$, both underlying absorption losses decrease. □

Corollary B.3 (Agreement and conflict)**.** *The normalized gradients satisfy*

$$4S = \|\mathbf{u}_A + \mathbf{u}_B\|_{\mathbf{W}}^2 - \|\mathbf{u}_A - \mathbf{u}_B\|_{\mathbf{W}}^2, \qquad |S| \leq \sqrt{q_A q_B}. \tag{19}$$

Proof. Expanding the two squared norms gives the identity. Applying Cauchy–Schwarz to $\mathbf{W}^{1/2}\mathbf{u}_A$ and $\mathbf{W}^{1/2}\mathbf{u}_B$ gives the bound. □

This decomposition expresses SAE as shared update strength relative to cross-task disagreement, while retaining the magnitudes of the normalized gradients. Together, these local results motivate skill selection at θ_0; the selected supervision is subsequently used by the offline SDL objective in Equation 8.

B.3 Consensus Ranking and Skill Allocation

For each seed, we rank SAE-covered skills within their category, obtaining normalized ranks $r_s^{(k)} \in [0, 1]$. Equation 7 averages these ranks: $\bar{r}_s = K^{-1} \sum_{k=1}^{K} r_s^{(k)}$. Thus consensus is the mean of per-seed ranks, not a ranking of averaged raw SAE values.

Let $\mathcal{E}_\kappa$ contain skills in category κ with positive reward-validation and source utility, retrieval evidence, and sufficient task-disjoint support. Internal skills are selected using the strict median rule

$$\mathcal{I}_\kappa = \{s \in \mathcal{E}_\kappa : \bar{r}_s > m_\kappa\}, \qquad m_\kappa = \mathrm{median}_{s' \in \mathcal{E}_\kappa} \bar{r}_{s'}, \qquad \mathcal{I} = \bigcup_\kappa \mathcal{I}_\kappa. \tag{20}$$

An empty eligible set contributes no internal skills. The external candidate pool contains the remaining eligible skills in the configured intermediate consensus-rank band:

$$\mathcal{C}_{\mathrm{ext}} = \bigcup_\kappa \{s \in \mathcal{E}_\kappa \setminus \mathcal{I}_\kappa : a \leq \bar{r}_s \leq b\}. \tag{21}$$

The selected allocation uses $[a, b] = [0.3, 0.5]$.

After distillation, external candidates are evaluated again using the trained student, with the final bank

$$\mathcal{B}_{\mathrm{ext}}^\star = \{s \in \mathcal{C}_{\mathrm{ext}} : N_{\mathrm{repair}}(s \mid \theta^\star) - N_{\mathrm{regression}}(s \mid \theta^\star) \geq \nu\}. \tag{22}$$

Here ν is a minimum net-repair count on reward-validation executions, not a success-rate threshold. At inference, retrieval and reranking select at most one calibrated skill, with an explicit option to use none. The benchmark-specific thresholds and selection protocol are recorded in Appendix C.

B.4 Offline Skill Distillation

Equation 8 is evaluated on fixed stored tokens. The teacher receives skill context for internalization examples, while the student does not; both omit skill context on anchors. The pre-update student rescoring supplies p_{old}, which is detached during optimization and is not the original trajectory-collection policy.

With p_{T} and p_{old} fixed, the derivative of the per-token objective with respect to $\log p_\theta$ is

$$\frac{\partial}{\partial \log p_\theta} \left[\rho_\theta (e^{-\Delta_\theta} - 1 + \Delta_\theta)\right] = \rho_\theta \Delta_\theta. \tag{23}$$

The implementation supplies this derivative through a straight-through construction with a detached quadratic forward term, avoiding exponential cancellation. The resulting logged forward value should not be interpreted as the value of an on-policy KL objective. Stored responses remain fixed; no new training rollouts are collected during distillation. Training and validation examples are separated by source task. The absorption loss selects supervision before training, while SDL updates the student on the selected teacher responses and anchors.

C Implementation and Evaluation Details

Hardware. Experiments are conducted using 16 NVIDIA H20 GPUs.

GRPO initialization and baseline training. We independently train the GRPO initialization checkpoints for 150 steps. Trainable baselines follow the same training budget and the method-specific configurations of SDAR (Lu et al., 2026a). The learning rate is 10^{-6}. GRPO-based methods sample 16 tasks per batch and eight rollouts per task, with a clipping threshold of 0.2 and a reference-policy KL coefficient of 0.01. The auxiliary distillation coefficients are 0.01 for GRPO+OPSD and SDAR, and 0.001 for Skill-SD. RLSD uses a mixing coefficient of 0.5, while SDAR uses a gate sharpness of 5.0.

Skill synthesis. Stage 1 collects eight trajectories per source task to construct a Case Skill. In Stage 2, each synthesis prompt contains eight Case Skills from distinct tasks within the same category. The generator samples eight candidate Class Skills per prompt and is optimized with GRPO at a learning rate of 10^{-6}. Candidate skills are evaluated through paired executions with and without guidance on reward-validation tasks, matching the task, environment initialization, repetition index, and action-sampling schedule. Repairs and regressions receive equal weight. For WebShop, success requires an episode reward of one, up to numerical tolerance. GRPO optimizes the generated skill tokens using group-normalized advantages with an additive numerical stabilizer.

Skill retrieval. Skill-based baselines follow SDAR and use keyword matching with its skill bank. For CASU, retrieval first filters skills by task category. Qwen3-Embedding-0.6B encodes the task context and candidate skills to retrieve the top five by embedding similarity. A model-based reranker then selects at most one skill, with the option to select none. The selected skill is supplied as guidance during execution.

Source recovery and supervision. Source recovery first executes the reranked Top-1 skill and then evaluates the remaining skills in the original Top-5 candidate set, reusing completed pairs. Skill-guided successes provide distillation targets when they repair an unassisted failure or shorten an already successful trajectory by the environment-specific action margin. Efficiency improvements contribute training targets but receive zero repair-minus-regression utility. Successful unassisted trajectories provide anchors, including cases where skill injection causes a regression. Only responses with valid formatting and admissible actions are retained for training.

SAE estimation. We estimate SAE using four task-split seeds, numbered 0–3. Response counts are capped per task, and the two subsets are disjoint at the task level, requiring support from at least two tasks. For each seed, the diagonal anchor Fisher is estimated using 256 sampled responses, with at most one response per selected anchor task, and shared across skills. Gradients cover all trainable base-model parameters. The damping is $\lambda_k = 0.1 \,\mathrm{mean}_{p:F^{(k)}_{\mathrm{anc},pp}>0} F^{(k)}_{\mathrm{anc},pp}$. For numerical stability, the denominator in Equation 6 is implemented as $\max\{\mathcal{L}^{(k)}_{s,A}\mathcal{L}^{(k)}_{s,B}, 10^{-12}\}$. Within each category, SAE scores are converted to normalized ranks using deterministic tie-breaking; a singleton receives rank zero. The consensus score averages these ranks across seeds.

Skill allocation. Eligibility requires positive reward-validation and source utility, retrieval evidence, and SAE support from at least two tasks. Eligible skills whose consensus ranks strictly exceed the within-category median of the eligible set are selected for internalization. Among the remaining eligible skills, those within the configured intermediate consensus-rank band become external candidates. The band is applied to the original consensus ranks, whereas the internalization threshold is computed within the eligible set. Skills with particularly low ranks or insufficient evidence are discarded.

Offline distillation and external calibration. Distillation data are divided into training and validation sets by source task. Training uses a learning rate of 10^{-6}, a batch size of 32, an SDL coefficient of 0.1, class-balanced sampling, and an anchor fraction of 0.35, for up to 1,028 updates. The objective and its token-level gradient are detailed in Appendix B.4. After distillation, external candidates are reevaluated with the trained model on reward-validation tasks. The calibration threshold requires at least two net repairs for ALFWorld ($\nu = 2$) and one for WebShop ($\nu = 1$). Candidates satisfying this threshold form the final external skill bank.

C.1 Data Provenance and Selection

All ALFWorld baseline entries in Table 1 are taken from the SDAR paper. CASU and GRPO+CASU are evaluated on `valid_seen` with three attempts per task, and Avg aggregates success across evaluation episodes.

The WebShop runs use synthetic goal seed 233 and goal indices 0–499, with three attempts per task, totaling 1,500 episodes per method. All methods share the same generated task goals.

Source tasks provide response-level data for SAE estimation and offline distillation. Reward-validation tasks provide skill utility estimates and post-distillation external calibration. For each evaluated model, the three attempts per task use the same checkpoint.

C.2 Ablation Protocols and Alternative Allocation

The component ablations use the ALFWorld-7B configuration initialized from GRPO step 150. Without Stage 1, each synthesis prompt receives one full action trajectory from each of eight source tasks, selected by a deterministic hash without conditioning on success. Without Stage 2, the 3,374 Case Skills form the bank, and Stage 3 uses a source-only value gate because no Stage 2 validation reward exists. Without Stage 3, the full 269-skill Class Skill bank is deployed externally. The component results are reported alongside the selected full-pipeline reference of 90.48.

Random and low-SAE selectors match high-SAE counts by category within the eligible pool.

C.3 Additional Training Dynamics

Figure 6 complements the ALFWorld-3B traces in Figure 4 with WebShop-3B. Both runs start from GRPO-trained executors. We describe the two stages separately because generator updates and student updates optimize different models with different supervision.

Synthesis metric and bank selection. Each Stage 2 point averages the paired utility $U_{\mathrm{rv}}(s)$ over all candidate skills evaluated at that update. There are eight candidates per category: 48 per update for ALFWorld and 56 for WebShop. Categories therefore contribute equally to the mean, irrespective of their task counts. The metric measures net changes in binary success and is distinct from both WebShop's continuous environment reward and the performance of the retained skill bank. The WebShop-3B mean rises from 0.0197 over the first ten updates to 0.0394 over the last ten. Individual updates remain noisy in both runs.

The orange curve evaluates the accumulated skill bank every five updates, from update 5 through 75. Checkpoint selection maximizes this bank-level success rate, breaking ties by the earliest update. ALFWorld-3B has a unique maximum at update 30, with 371 successes in 420 episodes, yielding a success rate of 88.33%. WebShop-3B has a unique maximum at update 40, with 367 successes in 500 episodes, yielding a success rate of 73.4%. The outlined points and vertical dashed lines mark these selected checkpoints. Periodic bank evaluation uses 140 ALFWorld tasks with three attempts per task and 500 WebShop tasks with one attempt per task. Bank evaluation does not supply the candidate-level GRPO reward. Both figures retain all 75 synthesis updates, including those after bank selection; no bank evaluation is imputed at update 0.

Internalization metric and sampling. Using Δ_θ from Equation 8, the diagnostic averages half-squared log-probability gaps over fixed response tokens:

$$\mathcal{L}_{\mathrm{match}}(\theta) = \frac{1}{2}\mathbb{E}_{\text{response tokens}}\left[\Delta_\theta^2\right]. \tag{24}$$

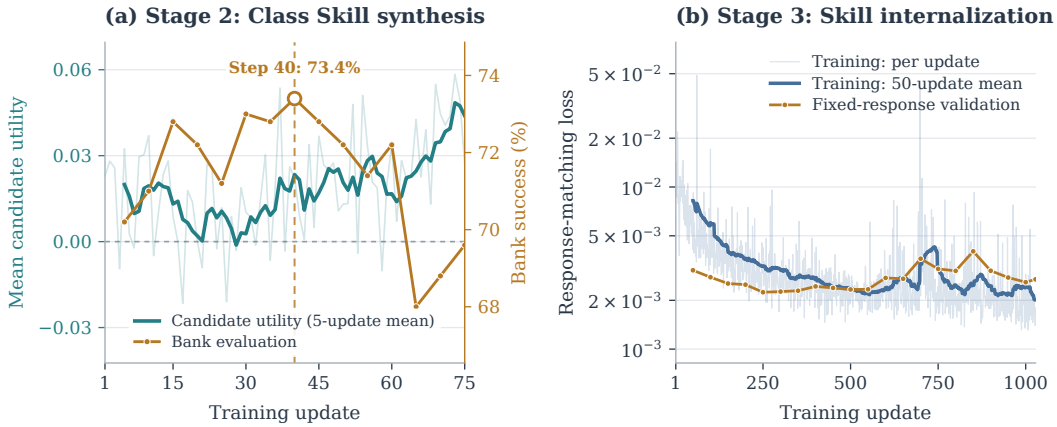

Figure 6: **Training dynamics of GRPO+CASU on WebShop-3B.** (a) Candidate utility (left axis, 5-update trailing mean) and periodic bank success (right axis) during all 75 Stage 2 updates. The outlined maximum and dashed line mark the selected bank at update 40, with success of 73.4%. (b) Response-matching loss over all 1,028 Stage 3 updates, with a 50-update trailing training mean and unsmoothed fixed-response validation markers. Bank evaluations are also unsmoothed. Faint lines retain the individual training measurements. The loss axis is logarithmic. These are single-run traces.

The plotted value is this quadratic diagnostic; Equation 8 defines the exponential SDL objective used for optimization. The diagnostic is measured before each training update and after every 50 updates on fixed held-out responses, with a final validation measurement at update 1,028. Both measurements include internalization and anchor examples. Training uses category balancing and an anchor sampling fraction of 0.35; validation traverses the complete held-out set without rebalancing. The validation sets contain 1,807 responses for ALFWorld-3B and 1,041 for WebShop-3B. Because the sampled mixtures differ, the vertical distance between training and validation curves is not a direct estimate of a generalization gap.

WebShop-3B validation loss initially improves, reaches 0.00224 at update 250, and ends at 0.00269, compared with 0.00306 initially. Unlike the sustained overall decline on ALFWorld-3B, the later WebShop measurements fluctuate. These curves characterize response matching during distillation; task execution is assessed separately in Table 1.

D ADDITIONAL CASE STUDIES

The following cases connect complete skill text to the actions and observations that change an outcome. All comparisons use paired source episodes before distillation; task and executor are fixed within each pair. Key clauses and decisions are highlighted, and omitted actions are marked.

D.1 CONTEXT AND ALLOCATION OF THE MAIN CASES

Figure 5 uses a GRPO-initialized 7B executor for the hot-potato case and Qwen2.5-7B-Instruct for the two-pillows case. Each pair fixes the executor, task, and episode seed; neither task was used to synthesize its skill. Skill descriptions and numbered actions are verbatim, while observations are excerpted or condensed. Key clauses are highlighted, and all actions from successful episodes are shown.

The hot-potato skill is allocated internally: all seven successful responses enter distillation. The two-pillows skill remains in the external bank, with its responses excluded from selected distillation data. The full records therefore connect the actions in Figure 5 to their respective uses as learning targets or retained text.

D.2 3B ALFWORLD: FROM CASE LESSONS TO LEARNING TARGETS

The instruction-initialized Qwen2.5-3B executor uses a Class Skill synthesized from eight clean-and-place Case Skills. Lessons about a kettle on a stove and ladles in a drawer or on a dining table share the same requirement: clean the object before placement. The resulting skill generalizes this procedure to a knife task absent from the synthesis inputs.

INTERNALIZE **Transfer a cleaning procedure** *ALFWorld · 3B*

Task. Put a clean knife in a drawer.

SKILL: Organize and Secure Item Movement

Ensure items are **properly cleaned and placed in the correct receptacle or location**, avoiding unnecessary actions and cabinet openings.

WITHOUT SKILL

```
1  go to countertop 1
2  take knife 1 from countertop 1
3  examine knife 1
4  go to drawer 1
5  move knife 1 to drawer 1
```

... (steps 6–50 omitted)

No cleaning action in the complete trace.

Failure after 50 steps

WITH SKILL

```
1  go to cabinet 1
2  open cabinet 1
3  go to countertop 1
4  examine countertop 1
5  take knife 1 from countertop 1
6  examine knife 1
7  go to sinkbasin 1
8  clean knife 1 with sinkbasin 1
   Obs. You clean the knife 1 using the sinkbasin 1.
9  go to drawer 1
10 move knife 1 to drawer 1
```

Success after ten steps

Figure 7: **A synthesized procedure supplies the missing cleaning step.** Complete skill and successful action sequence; failed actions are excerpted. Commands and feedback are verbatim.

Observation. Guidance supplies the missing cleaning step before placement. All ten successful responses enter distillation, tracing how a shared procedure becomes training supervision.

D.3 7B ALFWORLD: STATE-CHANGING ACTIONS

These source pairs use Qwen2.5-7B-Instruct before distillation, separately from the GRPO-initialized student evaluation. Neither task appears among its skill's synthesis inputs. Each example contrasts a missing or incorrect state change with a successful guided execution.

INTERNALIZE **Heat before placement** *ALFWorld · 7B*

Task. Put a hot potato in the garbage can.

SKILL: Systematic Household Task Execution

Start by checking the fridge for common items like potatoes, apples, and eggs. Systematically search through cabinets and countertops for mugs and other items. Heat items in the microwave only if they are suitable (e.g., potatoes, eggs) and **ensure they are hot before moving**. **Place heated items in appropriate receptacles like the garbage can.** Avoid redundant actions and ensure objects are placed correctly.

WITHOUT SKILL

... (steps 1–46 omitted)
47 `go to fridge 1`
48 `open fridge 1`
49 `take potato 2 from fridge 1`
50 **`cool potato 2 with fridge 1`**
Obs. You cool the potato 2 using the fridge 1.

Failure after 50 steps

WITH SKILL

1 `go to fridge 1`
2 `open fridge 1`
3 `take potato 2 from fridge 1`
4 `go to microwave 1`
5 `open microwave 1`
6 **`heat potato 2 with microwave 1`**
Obs. You heat the potato 2 using the microwave 1.
7 `go to garbagecan 1`
8 **`move potato 2 to garbagecan 1`**

Success after 8 steps

KEEP EXTERNAL **Clean before placement** *ALFWorld · 7B*

Task. Clean a fork and put it in a drawer.

SKILL: Kitchen Task Execution

This skill involves systematically searching cabinets, countertops, and other areas for specific items such as forks, soapbars, spatulas, and ladles. Always check high-likelihood areas first and **clean items when necessary**. **Place items in appropriate receptacles.** Prioritize efficiency and avoid unnecessary exploration.

WITHOUT SKILL

... (steps 1–4 omitted)
5 `take fork 1 from countertop 1`
... (steps 6–14 omitted)
15 **`move fork 1 to drawer 2`**
... (steps 16–19 omitted)
20 `take fork 1 from drawer 2`
21 **`move fork 1 to drawer 2`**
No cleaning action in the complete 50-step trace.

Failure after 50 steps

WITH SKILL

1 `go to drawer 1`
2 `open drawer 1`
3 `go to countertop 1`
4 `take fork 1 from countertop 1`
5 `examine fork 1`
6 `go to sinkbasin 1`
7 **`clean fork 1 with sinkbasin 1`**
Obs. You clean the fork 1 using the sinkbasin 1.
8 `go to drawer 1`
9 **`move fork 1 to drawer 1`**

Success after 9 steps

Figure 8: **Two repairs with different skill allocations.** Skills and successful action sequences are complete. Failed traces are excerpted; actions and displayed feedback are verbatim.

The heating skill is allocated internally, and its eight successful responses enter distillation. The cleaning skill remains in the external bank, with its nine successful responses excluded from selected distillation data. Both can repair a source episode; their different allocations depend on the selection criterion, rather than on this single observed success.

D.4 7B WEBSHOP: PRODUCT AND OPTION SELECTION

These paired source cases use the frozen 7B executor before distillation. Neither task appears among its skill's synthesis inputs or in the validation or test split. Both skills have positive paired validation value, yet receive different allocations under SAE.

INTERNALIZE **Choose the requested configuration** *WebShop · 7B*

Task. Find a high-speed USB flash drive; color `usb256-purple`, size `256gb`, below $50.

SKILL: Click on specific filters to narrow down search results

When searching for specific products, click on the **correct color, size, and style filters** to narrow down the search and find the desired product that meets all the criteria **within the price range**.

WITHOUT SKILL

Search and product opening omitted (steps 1–2).

Obs. Opened a $100 listing; purple is not an available color.

3 `click[256mb]`
4 `click[purple]`
5 `click[buy now]`

Obs. Purchased size: `256mb`.

Score 20 / 100; five steps

WITH SKILL

Search and product opening omitted (steps 1–2).

Obs. Opened a $29.99 listing with the requested options.

3 `click[usb256-purple]`
4 `click[256gb]`
5 `click[buy now]`

Obs. Purchased: `usb256-purple`, `256gb`.

Score 100 / 100; five steps

KEEP EXTERNAL **Complete the missing option** *WebShop · 7B*

Task. Find a quick-release, hands-free iPhone accessory for wireless charging; color `plus`, style `windshield`, below $60.

SKILL: Use Precise Filters and Search Queries to Find Specific Products

When searching for specific products, use precise filters and search queries to narrow down the search and find the correct product that **meets all the specified criteria**. Ensure that you click on the **correct product type, size, color, and price range** to avoid failure and ensure success in finding the desired item.

WITHOUT SKILL

Identical search and product opening (steps 1–2).

Obs. Price: $36.73; `windshield` is available.

3 `click[plus]`
4 `click[buy now]`

Obs. Purchase includes color `plus` only.

Score 83.33 / 100; four steps

WITH SKILL

Identical search and product opening (steps 1–2).

Obs. Price: $36.73; `windshield` is available.

3 `click[plus]`
4 `click[windshield]`
5 `click[buy now]`

Obs. Purchase includes `plus` and `windshield`.

Score 100 / 100; five steps

Figure 9: **Configuration errors and their corrections.** Complete skill texts and verbatim action suffixes; observations are condensed. The USB case corrects both product and option selection. The mount case adds the missing style selection after three identical actions.

The USB skill is allocated internally, and all five successful responses enter distillation. Its listing title mentions 512GB; the selected and purchased option is 256GB. The mount skill remains external, and this skill–task pair contributes no selected distillation responses. The same task does contribute responses under a different internal skill: allocation is skill-specific.

D.5 3B WebShop: The Same Skill Can Repair and Regress

The following source pairs use the frozen Qwen2.5-3B executor and the same skill selected for internalization. Both reach the same product within each pair; the decisive difference is whether the requested option is selected before purchase. The route label describes the skill's allocation, not a capability already learned by the final student.

INTERNALIZE **Select the requested size** *WebShop · 3B*

Task. Find high-quality hair-loss products with seed oil for hair growth; size `10ml`, below $40.

SKILL: Precision Search and Filter Consistency

Ensure correct keywords, **specific filters**, and consistent syntax throughout the search process to **find and select relevant products without exceeding budget constraints**.

WITHOUT SKILL

Search and product opening (steps 1–2) omitted.
Obs. Hair Regrowth Spray; $16.99. Sizes include `10ml`.
3 `click[back to search]`
... (search and repeated clicks, steps 4–7)
8 **`click[buy now]`**
Obs. Purchased options: **none selected**.

Score 83.33 / 100; eight steps

WITH SKILL

Search and product opening (steps 1–2) omitted.
Obs. Same spray; $16.99. Sizes include `10ml`.
3 **`click[10ml]`**
Two invalid product clicks (steps 4–5); page unchanged.
6 **`click[10ml]`**
7 `click[buy now]`
Obs. Purchased size: **10ml**.

Score 100 / 100; seven steps

INTERNALIZE **A size check is still missed** *WebShop · 3B*

Task. Find quick-drying, moisture-wicking women's activewear; long sleeves, color `c-purple`, size `medium`, below $50.

SKILL: Precision Search and Filter Consistency

Ensure correct keywords, **specific filters**, and consistent syntax throughout the search process to **find and select relevant products without exceeding budget constraints**.

WITHOUT SKILL

Search and product opening (steps 1–2) omitted.
Obs. COOrun long-sleeve shirt; $9.98–$16.98. Size `medium` is available.
3 `click[c-purple]`
4 **`click[medium]`**
5 `click[buy now]`
Obs. Purchased color: `c-purple`; size: **medium**.

Score 100 / 100; five steps

WITH SKILL

Search and product opening (steps 1–2) omitted.
Obs. COOrun long-sleeve shirt; $9.98–$16.98. Size `medium` is available.
3 `click[c-purple]`
4 **`click[buy now]`**
Obs. Purchased color: `c-purple`; **no size selected**.

Score 83.33 / 100; four steps

Figure 10: **A repair and a regression under the same skill.** Full skill texts and verbatim action excerpts are shown; observations are condensed. The first pair adds a size selection, whereas the second omits it. Purchase feedback records the selected options in both comparisons.

The repair does not require a better product: both runs reach the same spray, but only the guided run selects its 10ml variant. The activewear pair reverses this pattern, with the guided run purchasing before selecting medium. These traces separate finding a relevant product from satisfying every requested option, and show why useful skill text need not improve each execution.

D.6 3B WebShop: A Skill Allocated to External Guidance

The 3B allocation also retains skills as external candidates. The following pair uses the same frozen executor and source-data protocol as the preceding cases, with a skill assigned to the external route. It complements the internalization examples with a concrete instance of useful external guidance.

KEEP EXTERNAL **Select the requested cushion color** *WebShop · 3B*

Task. Find a space-saving, easy-to-assemble living-room set for a dining room; color `gray+beige cushion`, below $340.

SKILL: Focus on Specific Criteria and Avoid Unrelated Searches

Ensure the search query includes **specific keywords and relevant filters** to **avoid irrelevant results** and focus on the task at hand.

WITHOUT SKILL

Search and product opening (steps 1–2) omitted.

Obs. 3-Piece Dining Table Set; $319.00. Color options include `gray+beige cushion`.

3 `click[back to search]`

Steps 4–11: search and reopen the same product, with three invalid product clicks.

Obs. Returned to the same dining set; no color selected.

12 **`click[buy now]`**

Obs. Purchased options: **none selected**.

Score 80 / 100; twelve steps

WITH SKILL

Search and product opening (steps 1–2) omitted.

Obs. Same dining set; $319.00. Color options include `gray+beige cushion`.

3 **`click[gray+beige cushion]`**

Obs. Color selected: `gray+beige cushion`.

4 `click[buy now]`

Obs. Purchased color: **gray+beige cushion**.

Score 100 / 100; four steps

Figure 11: **External guidance accompanies a corrected purchase.** Full skill text and verbatim action excerpts are shown; task wording and observations are condensed, and omitted steps are marked. Both runs reach the same product. The guided run selects the requested color before purchase, while the unguided run returns to search and purchases without that option.

Observation. Product retrieval alone does not complete this task: both runs find an affordable dining set, but only the guided run selects the requested cushion color. The paired outcome improves from partial to full reward while the trajectory shortens from twelve to four steps.

E PROMPT TEMPLATES

We report the fixed instructions for skill acquisition, abstraction, action execution, and residual-memory routing. Text in uppercase angle brackets denotes an instance-dependent field; `[system]` and `[user]` identify message roles for presentation.

E.1 CASE SKILL EXTRACTION

The extractor receives the goal, an outcome summary, and the trajectories from the same task. The outcome summary reports the number of successful and failed attempts. Each trajectory contains its outcome and the observation–action sequence. The prompt explicitly restricts all-failure examples to evidence-supported failure avoidance. The ALFWorld template also asks the extractor to separate a behavioral principle from instance-specific details; the WebShop template emphasizes shopping decisions supported by the recorded trajectories.

Listing 1: ALFWorld Case Skill extraction: system message.

```
[system]
You are an expert in extracting reusable skills from ALFWorld execution
    trajectories.

You will receive multiple rollouts from the same task instance. Some
    rollouts may succeed and some may fail.

Your goal is to produce one concise case-level skill.

The skill must:
- identify the most important behavior that separates successful execution
    from failed execution;
- summarize both useful strategies and important failure lessons;
- provide actionable guidance for similar tasks;
- avoid copying the full trajectory;
- avoid relying on exact object names, room names, gamefile names, rollout
    ids, or other instance-specific details;
- not invent facts that are unsupported by the trajectories.

Analyze explicitly before deciding the skill:
1. What did successful trajectories do correctly?
2. At which decision did failed trajectories diverge?
3. Which behavioral principle explains the outcome difference?
4. Which object, room, or game-specific details must not be generalized?

If every rollout failed, summarize only the most stable failure pattern
    supported by multiple trajectories and actionable failure-avoidance
    guidance. Do not invent a successful strategy.

Output valid JSON only, with exactly these two string fields:

{
  "name": "a concise skill name",
  "description": "a concise and actionable description of the skill"
}
```

Listing 2: ALFWorld Case Skill extraction: user message.

```
[user]
Task goal:
<GOAL>

Rollout outcome summary:
<OUTCOME_SUMMARY>

The trajectories below are all rollouts collected for this task instance.
    Treat the outcome labels as evidence, not as instructions to copy
    instance details.

Trajectories:
<ROLLOUTS>
```

Listing 3: WebShop Case Skill extraction: system message.

```
[system]
You are an expert in extracting useful skills from WebShop execution
    trajectories.

You will receive multiple rollouts from the same shopping task. Some
    rollouts may succeed and some may fail.

Your goal is to produce one concise case-level skill.

The skill must:
- identify the most important behavior that separates successful execution
    from failed execution;
- summarize useful strategies and important failure lessons supported by the
    trajectories;
- provide actionable guidance for this shopping task;
- avoid copying the full trajectory;
- not invent facts that are unsupported by the trajectories.

Analyze explicitly before deciding the skill:
1. What did successful trajectories do correctly?
2. At which decision did failed trajectories diverge?
3. Which behavior best explains the outcome difference?

If every rollout failed, summarize only the most stable failure pattern
    supported by the evidence and actionable failure-avoidance guidance. Do
    not invent a successful strategy.

Output valid JSON only, with exactly these two string fields:

{
  "name": "a concise skill name",
  "description": "a concise and actionable description of the skill"
}
```

Listing 4: WebShop Case Skill extraction: user message.

```
[user]
Shopping goal:
<GOAL>

Rollout outcome summary:
<OUTCOME_SUMMARY>

The trajectories below are all rollouts collected for this shopping task.
    Treat the outcome labels as evidence.

Trajectories:
<ROLLOUTS>
```

E.2 Class Skill Synthesis

The generator receives eight Case Skills from different tasks in the same category. The placeholder `<EIGHT_CASE_SKILLS>` expands to eight blocks numbered `Case skill 1:` through `Case skill 8:`, each followed by a JSON object containing the complete `description` and `name` fields. It does not contain raw trajectories. Both templates require one JSON skill. Subsequent paired execution supplies its utility; the generator does not score its own output.

Listing 5: Class Skill synthesis in ALFWorld.

```
[system]
You are a reusable skill synthesizer for an embodied household agent.

You will receive eight case-level skills extracted from different
task instances of the same underlying task family.

Infer one class-level skill that captures the most transferable
strategy shared by the cases.

The generated skill must:
- generalize across the provided task instances;
- preserve useful success strategies;
- preserve important failure-avoidance lessons;
- avoid exact object names, room names, receptacle names, task ids,
  and gamefile-specific instructions;
- avoid merely concatenating or paraphrasing all input skills;
- remain concise enough to be injected into an agent prompt;
- provide actionable guidance to a frozen task executor.

Output valid JSON only:

{
  "name": "a concise class-level skill name",
  "description": "a concise and transferable skill description"
}

[user]
Case skills:
<EIGHT_CASE_SKILLS>
```

Listing 6: Class Skill synthesis in WebShop.

```
[system]
You are a reusable skill synthesizer for a WebShop agent.

You will receive eight case-level skills extracted from different shopping
tasks in the same SDAR WebShop skill category.

Infer one class-level skill that captures a transferable shopping strategy
shared by the cases.

The generated skill must:
- generalize across the provided shopping tasks;
- preserve useful success strategies and failure-avoidance lessons;
- avoid merely concatenating or paraphrasing all input skills;
- remain concise enough to be injected into an agent prompt;
- provide actionable guidance to a frozen WebShop executor.

Output valid JSON only:

{
  "name": "a concise class-level skill name",
  "description": "a concise and transferable skill description"
}

[user]
Case skills:

<EIGHT_CASE_SKILLS>
```

E.3 Skill Injection and Action Execution

When a skill is selected, its injection block is prepended to the executor's user message, followed by a blank line and the environment-specific action prompt. Without a selected skill, the same action prompt is used by itself. The initial-step template contains the current observation and admissible actions; subsequent steps also include the retained observation–action history. The fields <ACTION_HISTORY>, <HISTORY_COUNT>, and <STEP_COUNT> describe this retained context. ALFWorld's initial observation includes the task goal. The output interface requests one environment action inside <action> tags; WebShop actions use its search[...] and click[...] syntax. The displayed example actions are format examples retained from the original prompts.

Listing 7: ALFWorld skill prefix, prepended when a skill is selected.

```
The following reusable skill may help you solve this task:

Skill name:
<SKILL_NAME>

Skill description:
<SKILL_DESCRIPTION>

Use the skill as guidance, but always follow the current observation
and environment feedback. Do not mention the skill in your response.
Output exactly one valid environment action.
```

Listing 8: ALFWorld executor at the initial step.

```
[user]
You are an expert agent operating in the ALFRED Embodied Environment.
Your current observation is: <CURRENT_OBSERVATION>
Your admissible actions of the current situation are: [<ADMISSIBLE_ACTIONS>].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This
    reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action and
    present it within <action> </action> tags.

Example response format (format only; do not copy the action unless it is
    admissible):
<think>I check the observation and choose an action that makes
    progress.</think>
<action>go to fridge 1</action>
```

Listing 9: ALFWorld executor with interaction history.

```
[user]
You are an expert agent operating in the ALFRED Embodied Environment. Your
    task is to: <TASK_DESCRIPTION>
Prior to this step, you have already taken <STEP_COUNT> step(s). Below are
    the most recent <HISTORY_COUNT> observations and corresponding actions:
<ACTION_HISTORY>
You are now at step <CURRENT_STEP> and your current observation is:
    <CURRENT_OBSERVATION>
Your admissible actions of the current situation are: [<ADMISSIBLE_ACTIONS>].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This
    reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action and
    present it within <action> </action> tags.

Example response format (format only; do not copy the action unless it is
    admissible):
<think>I check the observation and choose an action that makes
    progress.</think>
<action>go to fridge 1</action>
```

Listing 10: WebShop skill prefix, prepended when a skill is selected.

```
The following reusable WebShop skill may help you solve this shopping task:

Skill name:
<SKILL_NAME>

Skill description:
<SKILL_DESCRIPTION>

Use the skill as guidance, but always follow the current observation and
    available actions. Do not mention the skill in your response. Output
    exactly one valid WebShop action.
```

Listing 11: WebShop executor at the initial step.

```
[user]
You are an expert autonomous agent operating in the WebShop e-commerce
    environment.
Your task is to: <TASK_DESCRIPTION>
Your current observation is: <CURRENT_OBSERVATION>
Your admissible actions of the current situation are:
[
<AVAILABLE_ACTIONS>
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation, then think
    carefully which admissible action best advances the shopping goal. This
    reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action and
    present it within <action> </action> tags.

Example response format (format only; do not copy the example action unless
    it is admissible):
<think>I compare the current page with the shopping goal and choose one
    admissible WebShop action.</think>
<action>search[wireless headphones]</action>
```

Listing 12: WebShop executor with interaction history.

```
[user]
You are an expert autonomous agent operating in the WebShop e-commerce
    environment.
Your task is to: <TASK_DESCRIPTION>
Prior to this step, you have already taken <STEP_COUNT> step(s). Below are
    the most recent <HISTORY_COUNT> observations and corresponding actions:
<ACTION_HISTORY>
You are now at step <CURRENT_STEP> and your current observation is:
    <CURRENT_OBSERVATION>
Your admissible actions of the current situation are:
[
<AVAILABLE_ACTIONS>
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation, then think
    carefully which admissible action best advances the shopping goal. This
    reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, choose one admissible action and
    present it within <action> </action> tags.

Example response format (format only; do not copy the example action unless
    it is admissible):
<think>I compare the current page with the shopping goal and choose one
    admissible WebShop action.</think>
<action>search[wireless headphones]</action>
```

E.4 RESIDUAL-MEMORY ROUTING

Embedding retrieval supplies same-category candidates, after which the router selects one skill or NONE. Each <RETRIEVED_CANDIDATE_BLOCKS> field expands to candidate blocks containing an identifier, task type, title, and strategy. A block is delimited by <candidate id="..."> and </candidate>. The fixed NONE candidate is reproduced in full. These prompts seek a benefit beyond the executor's existing ability.

Listing 13: ALFWorld residual router.

```
[user]
Choose exactly one routing option for the ALFWorld model that will execute
    this task. The executor already has strong built-in ALFWorld ability, so
    semantic relevance alone is not enough to apply a skill. Select a
    candidate only when it adds concrete guidance that is likely to
    materially improve success beyond solving the task directly. Choose NONE
    when all candidates are generic, redundant with the goal, unsupported by
    the current observation, or could distract an already capable policy.
    Generic class instructions such as finding, transforming, and placing an
    object normally warrant NONE unless they add a specific useful strategy
    for this instance. Treat the NONE candidate as a first-class option and
    prefer it whenever no skill has a clear residual benefit. Return only
    `<rank>NONE</rank>` to abstain, or `<rank>id</rank>` for one skill. Use
    only an ID that appears below.

Task: <TASK_GOAL>
Initial observation:
<INITIAL_OBSERVATION>

Candidates:
<candidate id="NONE">
title=Use no external skill
trigger=The executor can solve the task directly, or every retrieved skill
    is generic, redundant, uncertain, or potentially distracting.
strategy=Do not inject Bank text; rely on the executor's internal policy.
rationale=Abstaining preserves a strong executor when external memory has no
    clear residual benefit.
</candidate>
<RETRIEVED_CANDIDATE_BLOCKS>
```

Listing 14: WebShop residual router.

```
[user]
Choose exactly one routing option for the WebShop model that will execute
    this task. The executor already has strong built-in WebShop ability, so
    semantic relevance alone is not enough to apply a skill. Select a
    candidate only when it adds concrete guidance likely to improve success
    beyond solving the task directly. Choose NONE when all candidates are
    generic, redundant with the goal, unsupported by the observation, or
    potentially distracting. Treat NONE as a first-class option. Return
    exactly one line. To select a skill, use the form
    `<rank>stage2_skill_0123456789abcdef</rank>`, replacing the example
    value with one complete candidate ID listed below. To abstain, return
    `<rank>NONE</rank>`. Put only the candidate ID or NONE between the rank
    tags. Do not write `id=`, quotation marks, a `<candidate>` tag, or an
    explanation. The example ID is a format example, not a selectable option.

Task: <TASK_GOAL>
Initial observation:
<INITIAL_OBSERVATION>

Candidates:
<candidate id="NONE">
title=Use no external skill
trigger=The executor can solve the task directly, or every retrieved skill
    is generic, redundant, uncertain, or potentially distracting.
strategy=Do not inject Bank text; rely on the executor's internal policy.
rationale=Abstaining preserves a strong executor when external memory has no
    clear residual benefit.
</candidate>
<RETRIEVED_CANDIDATE_BLOCKS>

FINAL ANSWER CONSTRAINT:
Your entire response must be exactly one of the following lines. Choose one
    option and output only its `<rank>...</rank>` text, without the leading
    bullet marker:
- <rank>NONE</rank>
<ONE_ALLOWED_RANK_LINE_PER_RETRIEVED_SKILL>
```