DISCOVER AND INTERNALIZE: A CONTINUAL CO-EVOLVING TRAINING LOOP FOR SELF-IMPROVEMENT AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent self-improvement requires not only learning better behaviors, but also discovering what better behaviors could be. Existing approaches typically couple behavioral discovery with policy optimization, limiting the exploration of alternative behaviors and making it difficult to fully exploit potentially useful behavioral strategies. We reframe agent self-improvement as a behavioral hypothesis search problem and view skills as a behavioral hypothesis space: a skill is a candidate hypothesis to be tested, rather than an already validated capability. Based on this view, we propose Discover-and-Internalize Co-Evolution (DICE), a co-evolutionary framework between behavioral discovery and policy internalization. Behavioral discovery evolves and composes skills to explore new behavioral hypotheses, while policy internalization consolidates promising behaviors into the model weights. The updated policy then provides a stronger foundation for the next round of discovery, forming a mutually reinforcing evolution loop. Our results demonstrate that explicitly separating behavioral discovery from policy internalization, while combining them into a co-evolutionary loop, provides an effective paradigm for continual agent self-improvement.

1 INTRODUCTION

Reinforcement learning offers a natural route to self-evolving language model agents: by interacting with an environment and receiving feedback, an agent can refine its decisions and acquire more effective strategies for solving tasks. For agents that reason, use tools, and act over multiple turns, however, improvement requires more than reinforcing successful outcomes. An agent must also discover alternative ways of approaching a task and turn useful discoveries into capabilities that persist beyond the context in which they were found. These requirements point to two distinct processes: *behavioral discovery*, which identifies promising strategies, and *policy internalization*, which incorporates those strategies into the model's parameters.

Existing on-policy distillation and self-distillation approaches largely emphasize the second process. They transfer teacher behaviors into a student policy, using mechanisms such as skill distillation and step- or turn-level supervision to complement sparse outcome rewards. Much of their design effort concerns how to select, localize, or weight a teaching signal once a source of supervision has been constructed. Yet in agent self-improvement, the source itself need not be fixed: different skill configurations can induce different behaviors from the same underlying policy, and their relative effectiveness can vary across tasks and stages of training. Moreover, when the same model both executes a task and reflects on its own behavior, its judgments may reproduce the assumptions responsible for its failures. Improving the transmission of existing behaviors alone therefore does not resolve the question of how to discover and identify better behaviors to transmit.

We approach this question by viewing agent self-improvement as a search over behavioral hypotheses. In this view, a *Skill Repo* is not merely a memory of past successes or a library of reusable instructions; it is a *behavioral hypothesis space*. The term *skill* denotes a candidate prescription for how the agent might reason or act, not a capability that is assumed to be correct, general, or already mastered. Composing skills therefore creates new hypotheses about how an agent should behave. Such hypotheses must be evaluated rather than presumed beneficial: a skill-enhanced configuration

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.

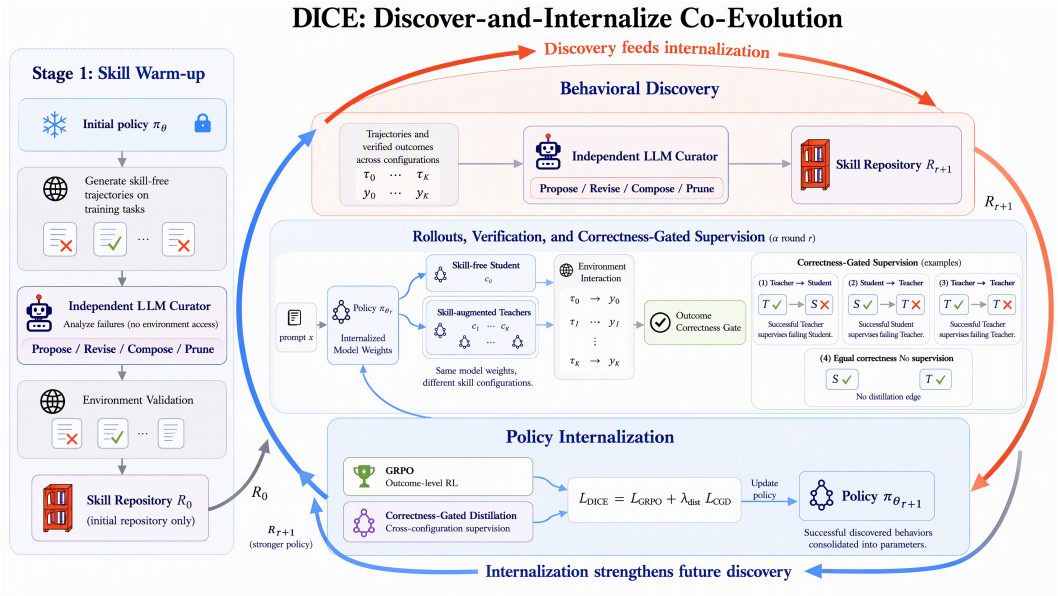


Figure 1: Overview of DICE. Skill warm-up initializes an external repository while the policy is frozen. During co-evolution, a skill-free Student and skill-augmented Teachers share policy weights, interact with the environment, and form correctness-gated supervision edges. Behavioral discovery updates the external hypothesis space, while GRPO and correctness-gated distillation internalize successful behaviors into the policy for the next round.

may uncover a successful strategy unavailable to the skill-free policy, but an unhelpful configuration may also interfere with behavior that the policy already performs correctly. This motivates a shift from assigning a permanent teacher to asking: *which configuration produces a successful behavior on the current task, and which configurations can learn from it?*

Based on this perspective, we propose *Discover-and-Internalize Co-Evolution* (DICE), a co-evolutionary framework that separates *behavioral discovery* from *policy internalization* while coupling them through iterative weight updates. The discovery process explores and composes behavioral hypotheses by evolving external skills, allowing the agent to test alternative strategies without requiring each candidate to be immediately encoded in its parameters. The internalization process then consolidates useful discovered behaviors into the model weights through distillation alongside outcome-based reinforcement learning. An independent LLM curator analyzes trajectories across configurations to inform behavioral comparison and skill evolution, while outcome correctness provides the gate for directional supervision. Once these behaviors are internalized, the updated policy becomes the foundation for the next round of skill-conditioned exploration and evaluation; newly discovered successful behaviors, in turn, provide supervision for subsequent policy updates. This reciprocal dependency closes the co-evolutionary loop between discovery and internalization.

The central mechanism is *correctness-gated three-way distillation*. For the same task prompt, a skill-free Student and two skill-enhanced Teachers, each using a different skill composition, generate their own trajectories. These names denote configurations rather than a fixed hierarchy of competence. Distillation proceeds from a configuration with a correct outcome toward one with an incorrect outcome: a successful Teacher can supervise a failing Student, a successful Student can reverse the direction and correct a failing Teacher, and the two Teachers can supervise one another when their outcomes disagree. Equal-correctness outcomes do not establish a direction under this gate. Thus, access to skills does not automatically confer teaching authority; the direction is determined by outcome correctness. The resulting behavioral supervision complements the outcome-level learning signal from Group Relative Policy Optimization (GRPO), using cross-configuration differences to expose opportunities for improvement that terminal rewards alone do not specify.

Our experiments examine how skill evolution and the granularity and direction of distillation shape behavioral discovery and policy internalization. DICE achieves state-of-the-art performance among models of comparable scale, demonstrating the effectiveness of co-evolving these two processes.

The results further reveal a discovery–internalization gap: evolving skills can improve the augmented Teacher rapidly, while the Student internalizes those improvements more gradually. This asymmetry highlights why stronger behavior under skill augmentation should not be equated with stronger intrinsic policy capability. Our central takeaway is therefore not simply that agents benefit from a stronger teacher, but that behavioral discovery and policy internalization should be allowed to evolve separately while remaining mutually reinforcing.

Our contributions are threefold:

- **The DICE framework.** We cast skills as behavioral hypotheses and introduce DICE, which couples external behavioral discovery with policy internalization in a continual training loop.
- **Correctness-gated cross-configuration distillation.** DICE uses verified outcomes to dynamically direct supervision among skill-free and skill-augmented configurations, enabling bidirectional and cross-Teacher transfer alongside GRPO.
- **Empirical analysis.** Experiments demonstrate strong performance and reveal a discovery–internalization gap, while showing that turn-level supervision improves behavioral transfer and that internalized skills persist without external augmentation.

2 Related Work

2.1 Agentic RL and On-Policy Self-Distillation

Agentic RL extends language-model post-training from single-turn generation to long-horizon interaction with tools and environments, where rewards are sparse, trajectories are long, and only a subset of actions contributes to outcomes. A central line of work therefore focuses on making training signals denser and more fine-grained.

GRPO Shao et al. (2024) removed the critic and estimated relative advantages from multiple rollouts of the same prompt, becoming the de facto standard for agentic RL. In parallel, on-policy distillation (OPD) was proposed to supervise students on their own rollouts, avoiding the distribution mismatch of offline distillation Agarwal et al. (2023). These directions converged into a joint outcome-RL plus OPD paradigm. Early instantiations introduced structural priors: Structured Agent Distillation Liu et al. (2025) applies segment-specific objectives over [REASON]/[ACT] spans, while SCoRe Kumar et al. (2024) trains multi-turn self-correction on the model's own error distribution. TRACE Tao et al. (2026) instead tackles turn-level credit assignment. Subsequent work replaced the external teacher with the student itself. SEED Wu et al. (2026) extracts hindsight skills from on-policy trajectories and distills them back into the policy as training-time context; OPID Yang et al. (2026b) derives hierarchical skills from the student's rollouts; StepOPSD Zhang et al. (2026b) shapes advantages from token log-probability differences at the agent-step level; TCOD Wang et al. (2026) and ATOD Tan et al. (2026) stabilize multi-turn distillation via short-to-long curricula and turn-aware annealing; DASH-OPD Xia et al. (2026b) switches the rollout executor by turn-level student–teacher discrepancy. More recent studies refine when and how strongly supervision should be applied: UCOB Tu et al. (2026) and BCSD Pan et al. (2026) compare multiple views of the same policy and select the better one as the local teacher; CRAFT Meng & Chen (2026) derives signed counterfactual token credit from GRPO's sibling rollouts; ICSD Lan et al. (2026) and OCSD Yang et al. (2026c) calibrate distillation weights by token influence or observation-ablated contrasts; SCOPE Zheng et al. (2026) routes supervision by trajectory correctness; RetireOPD Yu et al. (2026) retires the teacher adaptively; AOPD Jia et al. (2026) applies asymmetric objectives to positive and non-positive advantage regions.

2.2 Skill and Harness Evolution

A complementary line externalizes agent behavior into reusable artifacts. Voyager Wang et al. (2023) stores executable skills in an external library, and SkillWeaver Zheng et al. (2025) extends this to web agents; DreamCoder Ellis et al. (2020) provides a related program-induction view of reusable abstraction learning. Another strand evolves the agent harness itself: ADAS Hu et al. (2024), AgentSquare Shang et al. (2024), and MaAS Zhang et al. (2025a) search over agent designs and execution structures, while the Darwin Gödel Machine Zhang et al. (2025b) and AlphaEvolve AlphaEvolve Team (2025) evolve code and problem-solving systems.

Algorithm 1 Discover-and-Internalize Co-Evolution (DICE)

Require: Initial policy π_{θ_0}, training tasks $\mathcal{D}$, curator $\mathcal{C}$, rounds R, discovery width K
Ensure: Trained policy π_{θ_R} and skill repository $\mathcal{R}_R$

Freeze θ_0; collect trajectories; $\widehat{\mathcal{R}}_0 \leftarrow \textsc{Propose}(\mathcal{C}, \mathcal{D})$
$\mathcal{R}_0 \leftarrow \textsc{Validate}(\widehat{\mathcal{R}}_0, \pi_{\theta_0})$
for $r = 0, \ldots, R-1$ **do**
  $\mathcal{B}_r \leftarrow \emptyset$
  for all $x \in \mathcal{D}_r$ **do**
    $c_0 \leftarrow \emptyset$; sample/compose $c_{1:K} \subset \mathcal{C}(\mathcal{R}_r)$
    for $k = 0, \ldots, K$ **do**
      Roll out $\tau_k \sim \pi_{\theta_r}(\cdot \mid x, c_k)$ and verify outcome y_k
    end for
    Set $g_{i \to j} = \mathbb{I}[y_i > y_j]$; compute $\mathcal{L}_{\mathrm{GRPO}}$ and $\mathcal{L}_{\mathrm{CGD}}$
    $\mathcal{B}_r \leftarrow \mathcal{B}_r \cup \{x, c_k, \tau_k, y_k\}_{k=0}^{K}$
  end for
  $\theta_{r+1} \leftarrow \textsc{Optimize}(\theta_r, \mathcal{L}_{\mathrm{GRPO}} + \lambda_{\mathrm{dist}} \mathcal{L}_{\mathrm{CGD}})$
  $\widehat{\mathcal{R}}_{r+1} \leftarrow \textsc{CuratorUpdate}(\mathcal{C}, \mathcal{R}_r, \mathcal{B}_r)$
  $\mathcal{R}_{r+1} \leftarrow \textsc{Validate}(\widehat{\mathcal{R}}_{r+1}, \pi_{\theta_{r+1}})$
end for
return $\pi_{\theta_R}, \mathcal{R}_R$

More recent work studies policy–skill and policy–harness co-evolution: CoEvoSkills Zhang et al. (2026a) jointly evolves a skill generator and a surrogate verifier; SkillRL Xia et al. (2026a) maintains a hierarchical skill bank recursively updated with the policy; SIA Hebbar et al. (2026) and Co-Harness Chen et al. (2026) alternate between improving the harness and optimizing model weights; SkillForge Liu et al. (2026) organizes program synthesis around atomic skills with verifier-driven composition; SkillNet Liang et al. (2026) scales skill organization to hundreds of thousands of entries with multi-criteria evaluation. These methods mostly treat the policy and an external artifact as two entities that co-evolve.

3 Method

3.1 Overview

Behavioral discovery searches an external skill-based hypothesis space, while policy internalization consolidates successful behaviors into model weights. DICE alternates the two after first seeding the repository with validated skills. Figure 1 illustrates the architecture, and Algorithm 1 summarizes the training procedure.

Skill warm-up. DICE first freezes π_{θ_0} and lets the curator extract candidate strategies and error-avoidance rules from its trajectories. Environment validation retains only candidates that improve success, producing the seed repository $\mathcal{R}_0$ without updating policy weights.

Main loop. At round r, prompt x is rolled out under the skill-free configuration $c_0 = \emptyset$ and K retrieved or composed configurations $c_{1:K} \subset \mathcal{C}(\mathcal{R}_r)$, yielding trajectories $\tau_k \sim \pi_{\theta_r}(\cdot \mid x, c_k)$ and verified outcomes y_k. We call c_0 the Student and $c_{1:K}$ Teachers; these labels indicate skill access, not a fixed competence ordering.

Discovery compares these configurations and updates the repository through proposal, revision, composition, and pruning (Section 3.3). Correct outcomes define directional supervision (Section 3.2), which internalization combines with outcome-based RL. The repository therefore stores current hypotheses, whereas the weights accumulate validated behavior.

Training objective. The objective combines outcome RL with cross-configuration distillation:

$$\mathcal{L}_{\mathrm{DICE}} = \mathcal{L}_{\mathrm{GRPO}} + \lambda_{\mathrm{dist}}\, \mathcal{L}_{\mathrm{CGD}}, \tag{1}$$

where λ_{dist} controls distillation strength.

$\mathcal{L}_{\text{GRPO}}$ uses the standard group-normalized advantage over the $K+1$ verified rewards, clipped importance ratios, and a reference-policy KL penalty Shao et al. (2024). It supplies the outcome-level signal; Section 3.2 defines the dense cross-configuration term.

Loop closure. After each round, $\pi_{\theta_{r+1}}$ becomes the basis for new skill-conditioned exploration; new successful behaviors then supervise the next update. This reciprocal dependency closes the DICE loop.

3.2 INTERNALIZE: CROSS-CONFIGURATION DISCOVERY AND CORRECTNESS-GATED DISTILLATION

The $K+1$ configurations expose differences for skill evolution and form a dynamic graph that transfers successful behavior to unsuccessful configurations.

For binary outcome correctness, we define the directed gate between any two configurations $i, j \in \{0, 1, \ldots, K\}$ with $i \neq j$ as

$$g_{i \to j} = \mathbb{I}[y_i > y_j]\,. \tag{2}$$

An edge exists only when i succeeds and j fails. Applied to every pair, the gate yields:

- **Teacher → Student:** internalize a discovered strategy;
- **Student → Teacher:** preserve correct intrinsic behavior when a skill misleads;
- **Cross-Teacher:** exchange behavior across skill compositions.

Equal-correctness pairs create no edge; graded rewards use $g_{i \to j} = \mathbb{I}[R(y_i) > R(y_j)]$.

For an active edge, the successful configuration is evaluated on target j's on-policy states $h_{j,t}$:

$$\mathcal{L}^{\text{dist}}_{i \to j} = \sum_t w_{i,j,t}\, \mathcal{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid h_{j,t}, c_i) \;\Big\|\; \pi_\theta(\cdot \mid h_{j,t}, c_j)\Big)\,, \tag{3}$$

where $w_{i,j,t}$ selects supervised tokens, steps, or turns. This reveals how c_i would act at the decision points where c_j failed.

The full correctness-gated distillation objective aggregates over all active supervision edges:

$$\mathcal{L}_{\text{CGD}} = \frac{1}{Z} \sum_{\substack{i,j \in \{0,\ldots,K\} \\ i \neq j}} g_{i \to j}\, \mathcal{L}^{\text{dist}}_{i \to j}, \tag{4}$$

where Z is the number of active edges. Thus skills provide candidate sources, but observed outcomes grant teaching authority; augmentation never implies a fixed competence ordering.

3.3 DISCOVER: BEHAVIORAL HYPOTHESIS SPACE UPDATE

The repository $\mathcal{R}_r$ stores candidate textual strategies, decision rules, tool-use patterns, or error-avoidance principles. A configuration may retrieve one skill or compose several:

$$c = s_{i_1} \circ s_{i_2} \circ \cdots \circ s_{i_m}\,, \tag{5}$$

Membership does not certify usefulness; effects depend on the task and current policy and are verified through interaction.

At each round, the curator compares trajectories and proposes new skills, revisions, compositions, or pruning. Abstractly,

$$\mathcal{R}_{r+1} = \mathcal{U}\big(\mathcal{R}_r,\ \{x, c_k, \tau_k, y_k\}_{k=0}^{K}\big)\,, \tag{6}$$

where $\mathcal{U}$ combines curator proposals with environment-validated selection.

Revalidation is continuous: internalized skills may become redundant or harmful, while new failures create new hypotheses. Repository and policy updates therefore change one another's next evaluation conditions.

Table 1: Mean performance on ALFWorld and WebShop tasks. The best and second-best results within each model scale are shown in bold and underlined, respectively.

Model	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	Overall	Score	Success
Qwen3-1.7B	Vanilla	13.8	33.3	3.2	10.5	0.0	5.0	7.8	44.0	2.3
	GRPO	70.4	44.0	40.6	38.6	33.5	30.1	46.6	71.8	48.4
	OPSD	33.9	30.7	<u>54.1</u>	1.6	2.9	11.6	14.6	50.3	11.4
	GRPO+OPSD	63.1	41.5	39.2	44.8	27.3	29.1	41.7	72.1	49.0
	DICE (self-evolution)	70.5	45.5	30.0	40.1	22.9	<u>56.3</u>	44.2	<u>76.7</u>	52.1
	DICE (ours)	<u>73.9</u>	**69.7**	**58.8**	**53.5**	**48.9**	**63.6**	**61.4**	**87.5**	**69.5**
	RLSD	49.9	51.9	46.7	47.5	38.1	26.9	44.5	76.3	52.1
	SDAR	66.7	50.6	51.4	33.7	43.1	38.1	51.6	73.4	53.4
	OCSD	**75.9**	<u>63.1</u>	51.9	<u>47.7</u>	<u>46.4</u>	42.3	<u>55.5</u>	74.7	<u>54.4</u>
Qwen3-4B	Vanilla	60.0	33.3	16.7	31.6	0.0	20.0	27.3	16.2	3.9
	GRPO	86.6	59.5	77.7	67.1	55.5	55.0	70.6	81.9	69.5
	OPSD	69.9	39.2	26.6	26.6	19.0	26.4	36.2	27.8	4.7
	GRPO+OPSD	88.0	52.8	75.0	62.5	68.3	36.9	68.8	82.5	71.3
	DICE (self-evolution)	82.6	71.4	56.5	72.8	61.2	<u>69.5</u>	69.0	80.4	67.2
	DICE (ours)	**100.0**	**93.6**	**99.4**	**96.4**	**94.9**	**96.3**	**96.8**	**89.8**	**77.3**
	RLSD	87.9	<u>83.4</u>	80.5	70.3	79.6	60.6	79.2	83.7	68.5
	SDAR	85.2	63.9	76.8	80.0	72.9	42.6	73.2	83.6	71.9
	OCSD	<u>91.9</u>	77.2	<u>90.2</u>	<u>84.7</u>	<u>80.9</u>	63.8	<u>82.8</u>	<u>86.9</u>	<u>73.7</u>

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Tasks and evaluation. We evaluate on two complementary agent benchmarks. ALFWorld Shridhar et al. (2021) tests grounded household-task completion through text interaction, for which we report per-category and overall success. WebShop Yao et al. (2022) requires navigation and product selection under natural-language constraints; we report its environment score and task success rate.

Models and configurations. We use Qwen3-1.7B and Qwen3-4B Yang et al. (2025) to test whether the framework transfers across policy scales. Unless noted, evaluation uses the skill-free Student, so improvements measure capabilities stored in the model weights rather than inference-time retrieval. Section 4.3 uses ALFWorld and Qwen3-4B. The curator only analyzes cross-configuration trajectories; it never answers tasks, and verified outcomes—not curator judgment—determine distillation direction.

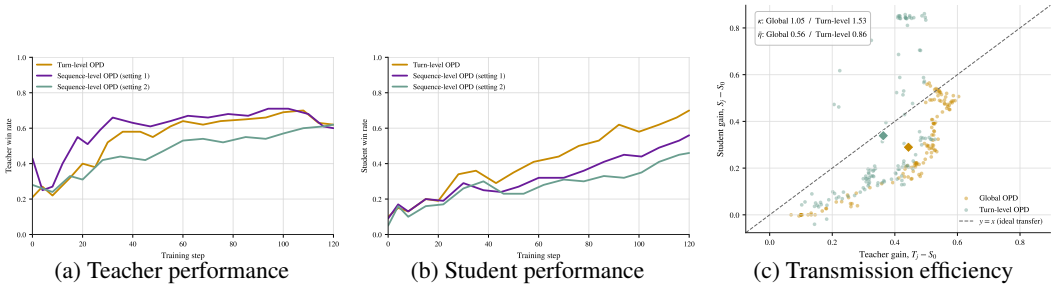
4.2 MAIN RESULTS

Baselines. Table 1 compares Vanilla, GRPO, OPSD, GRPO+OPSD, and a self-evolution variant in which the policy is both actor and analyzer. We also include RLSD Yang et al. (2026a), SDAR Lu et al. (2026), and OCSD Yang et al. (2026c). All methods at each scale share the initial checkpoint.

Results. DICE is strongest overall at both scales. At 1.7B it leads ALFWorld overall by 5.9 points and the second-best WebShop score and success rate by 10.8 and 15.1 points; at 4B it leads every reported metric, including ALFWorld overall by 14.0 points. Gains across acquisition, transformation, and WebShop tasks show that the benefit extends beyond exploration or a single action space. The self-evolution variant is consistently weaker, supporting the separation of execution from independent behavioral analysis. Since evaluation is skill-free, the gains reflect internalized policy capability rather than inference-time repository access.

4.3 ABLATIONS AND ANALYSIS

Teacher-to-Student transmission efficiency. Stronger augmented behavior is useful only if it can be transferred to the skill-free policy. We therefore compare global and turn-level OPD with matched Teachers, initialization, data, rollout budget, and optimization steps. Global OPD supervises the first 256 tokens of the complete action sequence, whereas turn-level OPD resets a 64-token supervision



(a) Teacher performance (b) Student performance (c) Transmission efficiency

Figure 2: Teacher-to-Student transmission under global and turn-level OPD. Panels (a) and (b) compare the win-rate trajectories of the skill-conditioned Teachers and their resulting Students. Panel (c) pairs each available Teacher gain with the Student gain obtained from it; the dashed line indicates ideal one-to-one transfer. Turn-level OPD transfers a larger fraction of the available gain and better preserves differences among Teacher configurations.

window at each interaction turn. This isolates loss support from Teacher construction and training compute.

For Teacher SR T_j, initial Student SR S_0, and distilled Student SR S_j^ℓ under loss ℓ, we measure the fraction of available Teacher gain transferred and the fraction of cross-Teacher variation preserved:

$$\eta_j^\ell = \frac{S_j^\ell - S_0}{\max(T_j - S_0, \epsilon)}, \tag{7}$$

$$\kappa^\ell = \frac{\max_j S_j^\ell - \min_j S_j^\ell}{\max_j T_j - \min_j T_j}. \tag{8}$$

Figure 2 shows substantial variation among available Teachers. Global OPD attenuates these differences after transfer, producing clustered Student curves; changing only to turn-level support yields a steeper and higher learning curve. At the gain level, mean transfer rises from 0.56 to 0.86, while the preservation ratio rises from 1.05 to 1.53 and points move closer to ideal one-to-one transfer. Thus Teacher quality alone does not determine Student improvement: global aggregation discards behavioral distinctions that turn-localized supervision converts into policy capability. This transfer bottleneck helps explain the discovery–internalization gap below.

Discovery–internalization dynamics. At round r, let S_r be skill-free Student SR before internalization and T_r the best augmented Teacher SR under the current repository. We report

$$G_r^{\text{disc}} = |T_r - S_r|, \qquad G_r^{\text{int}} = S_{r+1} - S_r, \qquad R_r^{\text{int}} = \frac{G_r^{\text{int}}}{\max(G_r^{\text{disc}}, \epsilon)}. \tag{9}$$

G_r^{disc} is an absolute disparity rather than a signed Teacher advantage because correctness gating can reverse supervision. Teachers lead during the early rounds ($\text{AUC}_T = 0.662$ versus $\text{AUC}_S = 0.437$), but the Student overtakes them by round 16. Crucially, the absolute disparity does not vanish: the repository continues to induce behaviors different from the skill-free policy, so it remains a source of supervision even after the conventional Teacher advantage reverses. Mean internalization rate is only 0.353 (median 0.132) and remains below one, showing that each update absorbs only part of the behavioral variation exposed in that round. Repository operations can reshape the hypothesis space at once, whereas distillation changes the policy incrementally at individual decision points. The later reversal further indicates policy dependence: once a strategy is internalized, its originating skill may become redundant or harmful, motivating repeated validation and pruning. This asymmetry also explains why Teacher SR alone is an incomplete progress measure. The augmented frontier records behaviors currently exposed by the repository, whereas skill-free SR measures durable policy capability. Their gap is therefore both a training signal and a diagnostic of internalization bandwidth, rather than simply evidence of a weak Student or a strong Teacher.

Ablations on DICE. Under a matched protocol, we independently enable repository evolution and distillation-based internalization; the single-component variants use either evolving skills without distillation or a fixed repository with distillation. Table 2 reports skill-free Student SR. Their non-additive interaction is

$$\Delta_{\text{coev}} = P_{\text{full}} - P_{\text{disc}} - P_{\text{int}} + P_{\text{neither}} \tag{10}$$

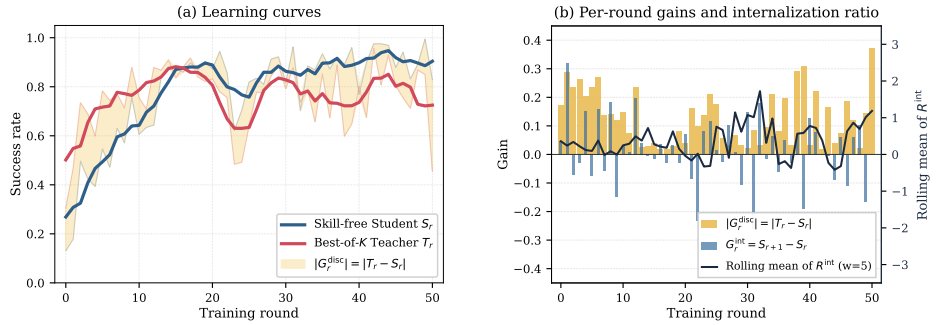


Figure 3: Discovery–internalization dynamics across training rounds, showing persistent behavioral disparity and incomplete per-round transfer.

Table 2: Core ablations on ALFWorld with Qwen3-4B. SR is measured on the skill-free Student. Upper rows ablate discovery and internalization; lower rows retain both but restrict distillation direction.

Variant	Discovery	Internalization	Distillation direction	SR
GRPO only (P_{neither})	✗	✗	None	31.5
Discovery only (P_{disc})	✓	✗	None	60.1
Internalization only (P_{int})	✗	✓	Standard three-way	49.2
Full DICE (P_{full})	✓	✓	Standard three-way	**96.8**
Unconditional T→S	✓	✓	T→S	38.6
Bidirectional S↔T only	✓	✓	S↔T only	42.1

We additionally restrict distillation directions under matched rollout and token budgets.

Discovery and internalization alone reach 60.1 and 49.2 SR, whereas full DICE reaches 96.8, yielding $\Delta_{\text{coev}} = 19.0$. This positive interaction supports reciprocal feedback: policy updates change which skills remain useful, and repository updates refresh the behaviors available for learning. Direction is also critical: unconditional T→S and gated S↔T reach only 38.6 and 42.1, showing that outcome-directed Student and cross-Teacher edges prevent harmful one-way transfer and propagate complementary discoveries.

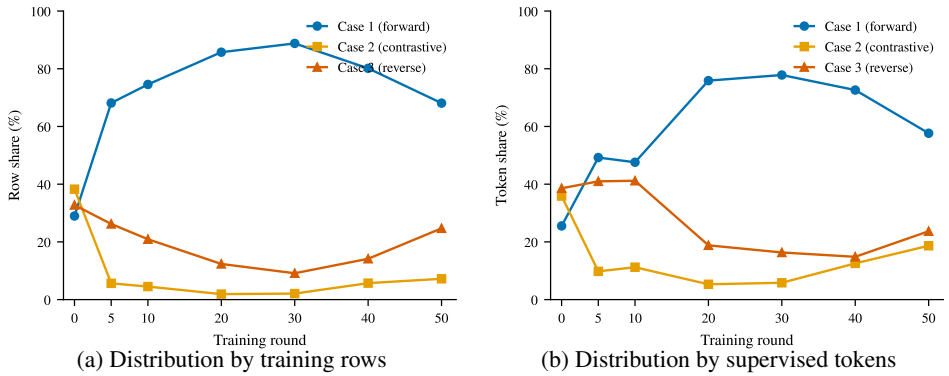
Distillation direction analysis. Figure 4 shows an early rise and later decline in Teacher-to-Student signals. Case 1 grows from 237 accepted rows at round 0 to 812 at round 20 before declining, while corrective Case 3 falls and recovers after round 30. Thus, the repository initially supplies behaviors for Student acquisition, but later helps select, correct, or reject skills as the Student improves. Row- and token-level views agree, ruling out trajectory length as the explanation. The return of corrective edges further shows that hypotheses useful to an earlier policy can become mismatched after internalization; correctness gating turns this non-stationarity into useful supervision.

Case study: skill internalization. Table 3 compares behavior before and after internalization in two tasks. Table 4 then gives the skill content associated with Case 1.

Case 1 tests completeness: at a new object location, the skill-free final Student preserves all four clauses—directed search, exact acquisition, fridge-opening before cooling, and termination after placement. Case 2 tests selectivity: it retains the mandatory take–clean–place protocol but removes the Teacher's wrong-cup detour. Agreement on required clauses alongside disagreement on incidental actions supports strategy internalization rather than trajectory memorization.

5 Conclusion

We presented *Discover-and-Internalize Co-Evolution* (DICE), which treats external skills as behavioral hypotheses and couples their outcome-grounded evolution with policy internalization. Correctness-gated three-way distillation dynamically transfers behavior among skill-free and skill-augmented configurations without assuming a fixed teacher hierarchy. Experiments and ablations



(a) Distribution by training rows (b) Distribution by supervised tokens

Figure 4: Distribution of correctness-gated distillation directions over training, measured by (a) accepted rows and (b) supervised tokens. Both views reveal a shift from early Teacher-to-Student transfer toward later selective correction.

Table 3: Skill-internalization cases comparing failed, augmented, and internalized behavior.

Case 1: Cool and place a pan on the dining table			
Phase	Student FAIL (ckpt 179, 50 steps, no skill)	Teacher WIN (rid 192, 8 steps, skill)	Student SUCCESS (ckpt 329, 13 steps, no skill)
Search & navigate	steps 1–24: wanders across counters, dining table, fridge, shelves, and microwave; no pan.	steps 1–2: countertop 1 → countertop 2 (skill-directed).	steps 1–7: countertop 1 → countertop 2 → diningtable → stoveburner 1–4 (open-surface sweep, no revisit).
Take	step 25: `take pot 1 from shelf 2` (pot ≠ pan).	step 3: `take pan 3 from countertop 2` — exact target.	step 8: `take pan 1 from stoveburner 4` — exact target at a *new* location.
Cool	step 27: `cool pot 1 with fridge 1` — cools the wrong object.	step 6: `cool pan 3 with fridge 1`; fridge opened first.	step 11: `cool pan 1 with fridge 1` — correct after `go to fridge 1` + `open fridge 1`.
Place	step 29: `move pot 1 to diningtable 1` — wrong object ⇒ task fails.	step 8: `move pan 3 to diningtable 1`; wins.	step 13: `move pan 1 to diningtable 1`; wins.
Aftermath	steps 30–50: `look`/`inventory` and location loop until budget exhaustion.	8 steps total.	13 steps: 7 search + 6 protocol; stops immediately.
Case 2: Clean and place a fork on the countertop			
Phase	Student FAIL (ckpt 29, 17 steps, no skill)	Teacher WIN (rid 42, 35 steps, skill)	Student SUCCESS (ckpt 119, 7 steps)
Search	steps 1–15: searches many containers; reaches dining table late.	steps 1–16: exhaustive search through fridge, counter, cabinets.	steps 1–2: countertop 1 → diningtable 1 (2 steps direct)
Take	step 16: `take fork 1 from diningtable 1` — correct but too late	step 17: `take cup 2`; step 31: `take fork 1` (corrects detour)	step 3: `take fork 1 from diningtable 1` — correct immediately
Clean	never cleans — root cause of failure	step 19: `clean cup 2`; step 33: `clean fork 1 with sinkbasin 1`	step 5: `clean fork 1 with sinkbasin 1`
Place	step 17: `move fork 1` (uncleaned → fails)	step 21: `move cup 2`; step 35: `move fork 1 to countertop 1`	step 7: `move fork 1 to countertop 1` — wins
Result	skips clean, fails at 17 steps	35 steps (incl. wrong-take detour), wins	7 steps, wins; faster than teacher

Table 4: Content of the `cool_and_place` skill used in Case 1.

Clause	**Behavioral prescription**	**Clause**	**Behavioral prescription**
Directed search	Begin with `go to`; prioritize open surfaces and avoid revisits.	Exact acquisition	Verify the requested type and take it immediately.
Cooling protocol	Navigate to and open the fridge before cooling.	Exact completion	Place the correct object and stop after success.

show that discovery and internalization reinforce one another, while their different learning rates create a persistent gap between behaviors exposed by skills and capabilities absorbed into model weights. These findings support separating the two processes while optimizing them through a reciprocal loop, providing a general framework for continual agent self-improvement.

AI Use Statement. Generative AI tools were used to assist with language editing, formatting, and presentation. The authors reviewed and verified all AI-assisted content and remain fully responsible for the paper's claims, experiments, analysis, and conclusions. This writing assistance is separate from the use of LLM curators as an explicit component of the DICE method described in the paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. *arXiv preprint arXiv:2306.13649*, 2023.

AlphaEvolve Team. Alphaevolve: A coding agent for scientific and mathematical discovery. *Nature*, 2025. To appear; DOI: 10.1038/s41586-025-09085-4 (verify).

Zhengyu Chen, Teng Xiao, Huaisheng Zhu, Yige Yuan, Luan Zhang, and Jingang Wang. Co-harness: Co-evolving harnesses and model weights for llm agents. *arXiv preprint arXiv:2607.22688*, 2026.

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.

Prannay Hebbar, Yogendra Manawat, Samuel Verboomen, Alesia Ivanova, Selvam Palanimalai, Kunal Bhatia, et al. SIA: Self improving ai with harness & weight updates. *arXiv preprint arXiv:2605.27276*, 2026.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*, 2024.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, et al. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Katherina Baumgartner, Sharmin Hu, Joy Gu, Alexander Ibarz, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Qizhen Lan, Xi Xiao, Xiangchen Guan, Mengchen Fan, Moule Lin, Jung Im Choi, et al. Trust is not enough: Influence calibration for on-policy self-distillation in agentic rl. *arXiv preprint arXiv:2608.14945*, 2026.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, et al. SkillNet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*, 2026.

Jun Liu, Zhenglun Kong, Peiyan Dong, Changdi Yang, Tianqi Li, Hao Tang, Puwei Ma, Chengzhong Xu, Xin Yuan, Yanzhi Chen, et al. Structured agent distillation for large language model. *arXiv preprint arXiv:2505.13820*, 2025.

Yanming Liu, Xinyue Peng, Jiannan Cao, Xinyi Wang, and Jinbo Su. SkillForge: Compositional skill synthesis with verification-in-the-loop for generating formally verified dafny programs. *arXiv preprint arXiv:2608.29841*, 2026.

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*, 2026.

Zibin Meng and Kani Chen. CRAFT: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*, 2026.

Tianjun Pan, Yuan Li, Hongda Wang, Linbo Jin, Mengfei Song, Lei Gao, et al. Bidirectional context self-distillation for reinforcement learning of skill-based llm agents. *arXiv preprint arXiv:2608.09555*, 2026.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, et al. AgentSquare: Automatic llm agent search in modular design space. *arXiv preprint arXiv:2410.06153*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haoyu Zhang, Mingchuan Zhang, Y.K. Li, Yihan Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. ATOD: Annealed turn-aware on-policy distillation for multi-turn agentic tasks. *arXiv preprint arXiv:2606.27814*, 2026.

Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, et al. TRACE: Turn-level reward assignment via credit estimation for long-horizon agents. *arXiv preprint arXiv:2607.13988*, 2026.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, et al. UCOB: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026.

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. *arXiv preprint arXiv:2305.16291*, 2023.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. TCOD: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. *arXiv preprint arXiv:2604.24005*, 2026.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Chenyang Gong, et al. SEED: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, et al. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026a.

Yuchen Xia, Qianguo Sun, Chao Song, Junlong Wu, Yiyan Qi, and Yunjian Xu. DASH-OPD: Discrepancy-aware switching with hysteresis for on-policy distillation. *arXiv preprint arXiv:2607.29078*, 2026b.

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

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.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, et al. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026b.

Yi Yang, Cong Qin, Xiaodan Liu, Chishui Chen, Qing Dong, Yan Zhang, et al. Agentic reinforcement learning with observation-calibrated self-distillation. *arXiv preprint arXiv:2608.04788*, 2026c.

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

Yan Yu, Zhengxi Lu, Yizhou Liu, Yichen Pan, Aozhe Wang, Qipeng Chen, et al. Retire-OPD: Self-retiring on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2609.20784*, 2026.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025a. ICML 2025 Oral.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, et al. CoEvoSkills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*, 2026a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025b.

Yanfei Zhang, Xu Lin, and Chenglin Wu. StepOPSD: Step-aware online preference self-distillation for agent reinforcement learning. *arXiv preprint arXiv:2605.27140*, 2026b.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, et al. SCOPE: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, and Xinyun Chen. SkillWeaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

A Additional 8B-Scale Evaluation

Due to the computational and time constraints of the submission schedule, we were unable to complete the full set of experiments with an 8B-parameter policy before submission. Consequently, the empirical claims in the main paper are based only on the model scales for which all reported experiments were completed. We commit to completing the corresponding 8B experiments during the rebuttal period and reporting the results in the rebuttal and the final version of the paper. These additional experiments will follow the same training protocol, evaluation settings, and skill-free evaluation criteria as the experiments reported in Section 4. If the paper is accepted, we will immediately release a public code repository containing the DICE training and evaluation implementation, configuration files, skill-evolution components, and reproduction scripts.

B Experimental and Methodological Details

This appendix provides implementation details for the ALFWorld experiments. We use the same Student, Teacher, curator, skill repository, verified outcomes, and DICE objective defined in Section 3.1; no additional training objective or configuration type is introduced here.

B.1 Training and Evaluation Protocol

The ALFWorld training split contains 3,553 games. Each interaction is limited to 50 environment steps, and the policy observes the three most recent turns. For every task, we generate one skill-free Student configuration and two skill-augmented Teacher configurations, corresponding to $K = 2$ in Section 3.1. Each Teacher receives a composition of two skills retrieved from the current repository. Rollouts use temperature 1.0, and eight samples are generated per prompt for the GRPO group.

Evaluation uses at most 140 games from the ALFWorld `valid_seen` split. Unless an analysis explicitly compares configurations, all reported results use the skill-free Student. The evaluation therefore measures behavior internalized in policy weights rather than gains from inference-time skill augmentation.

B.2 Policy Optimization

We optimize the DICE objective defined in Section 3.1. The GRPO term uses the verified environment outcomes of the Student and Teachers. The correctness-gated distillation term is evaluated only on directed edges whose source succeeds and target fails, exactly as specified in Section 3.2. We use turn-level supervision because the ablation in Section 4.3 shows that it transfers multi-step behavior more effectively than coarser trajectory-level supervision.

Optimization uses Adam with a constant learning rate of 10^{-6}, weight decay 0.1, and momentum coefficients 0.9 and 0.98. The distillation weight λ_{dist} is 0.4. GRPO uses clipping values 0.2 and

0.28, a reference-policy KL coefficient of 0.001, and an entropy coefficient of 0.01. Gradients are accumulated in FP32, while model forward and backward passes use mixed precision.

B.3 Curator and Repository Update

The curator compares successful and unsuccessful trajectories produced for the same task or task type. It summarizes the behavioral difference, proposes a new skill or a revision to an existing skill, and may recommend composing or removing repository entries. Candidate edits are expressed as textual strategies or action rules; the curator does not assign distillation direction and never acts in the environment. Teaching authority remains determined only by verified outcomes.

We use GLM-5.2 and Gemini 3.1 Pro as curators. Repository updates follow the operations described in Section 3.3: proposal, revision, composition, and pruning. The updated repository is used by the Teachers in the next round. We cap the repository at 30 skills and remove the least useful entries when the cap is exceeded.

B.4 Stability and Data Filtering

Distillation is applied to at most the first 256 response tokens of each turn, where skill-conditioned differences primarily affect planning and action selection. Tokens later in the same turn receive only the GRPO signal. This restriction prevents response length from dominating the distillation term.

We additionally penalize responses longer than 8,192 tokens with coefficient 10^{-4} and keep rewards within the interval $[0, 1]$. Responses shorter than 50 tokens and responses whose non-ASCII character ratio exceeds 0.3 are excluded from optimization because they usually correspond to empty or malformed generations. These filters do not alter environment success labels.

B.5 Hyperparameter Summary

Group	Hyperparameter	Value
Environment	train / evaluation games	3,553 / ≤ 140
	maximum interaction steps	50
	observation history	3 turns
DICE rollout	discovery width K	2
	skills per Teacher	2
	samples per prompt	8
	rollout temperature	1.0
Filtering	supervised tokens per turn	256
	length threshold / penalty coefficient	8192 / 10^{-4}
	minimum length / maximum non-ASCII ratio	50 / 0.3
DICE objective	GRPO advantage	group-normalized
	distillation weight λ_{dist}	0.4
	reference-policy KL / entropy coefficients	0.001 / 0.01
	clipping values	0.2 / 0.28
Optimizer	learning rate / schedule / weight decay	10^{-6} / constant / 0.1
	Adam momentum coefficients	0.9 / 0.98
	gradient accumulation precision	FP32
Skill evolution	update method	contrastive trajectory comparison
	Curator	GLM-5.2 / Gemini 3.1 Pro
	maximum repository size	30 skills
System	training / rollout GPUs	2 / 6
	rollout-engine concurrency	64
	maximum tokens per GPU	16,384