GET: DIRECT PEER EVIDENCE FOR AGENTIC SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome-supervised reinforcement learning provides language agents with limited guidance for intermediate decisions. On-policy self-distillation (OPSD) supplies token-level supervision from privileged context, making the selection and organization of decision-relevant experience a central design question. We introduce Group Evidence Teacher (GET), which organizes observation-matched peer experience into a decision-local reference alongside immediate feedback on the student's own action. For each decision, GET selects transitions from other rollouts of the same task and deterministically compiles their actions and source episodes' outcomes into an Evidence Card, without generating skills or corrective guidance. During training, the same policy re-scores the student-sampled response using a decision-local context built from available action feedback and peer evidence; the resulting token-level log-probability differences determine the weights of an auxiliary update alongside group-relative reinforcement learning. On ALFWorld, GET achieves 77.25% mean success across three seeds. Adding peer evidence to action feedback yields a 7.66-percentage-point gain in task success in a seed-0 component comparison; environment-verified replay of these policies shows a 13.29-point gain in required object-state changes across 143 cleaning, heating, and cooling tasks. On Search-QA, GET outperforms Skill in both micro- and macro-averaged exact match across both backbones and weighting rules. With Qwen3-1.7B and Raw OPSD, GET achieves higher test accuracy with 23.0% fewer search calls per question than Skill.

1 INTRODUCTION

Outcome-based reinforcement learning trains language agents from the success or failure of complete interactions (Wang et al., 2025). With terminal rewards, Group Relative Policy Optimization (GRPO) assigns a shared outcome-based advantage to the generated tokens within each trajectory, offering limited differentiation among the individual decisions that produced the result (Shao et al., 2024; Feng et al., 2025). This coarse granularity becomes especially consequential in multi-turn tasks, where an agent must assess new observations, revise its plan, and recover from unsuccessful actions (Yao et al., 2023). Learning these behaviors calls for finer-grained supervision that reflects the decisions the agent actually makes.

On-policy self-distillation (OPSD) supplies complementary token-level supervision by letting the policy learn from a more informed view of its own responses (Zhao et al., 2026; Wang et al., 2026; Lu et al., 2026). The student acts from its ordinary interaction history, while the same policy, conditioned on privileged training-time context, serves as its teacher. Scoring the student's sampled response under this richer context turns additional information into a learning signal on the student's own behavior. The value of this supervision therefore depends on what the teacher knows about the decision being evaluated.

Existing methods construct privileged context from procedural skills, successful and failed peer solutions, or search traces (Wang et al., 2026; Lu et al., 2026; Yu et al., 2026; Ma et al., 2026). In multi-turn interaction, attempts at the same task can reach different situations, so task identity alone does not determine which peer experience is useful for a given decision. We study how to organize records already collected within a rollout group into a reference for each sampled decision.

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.

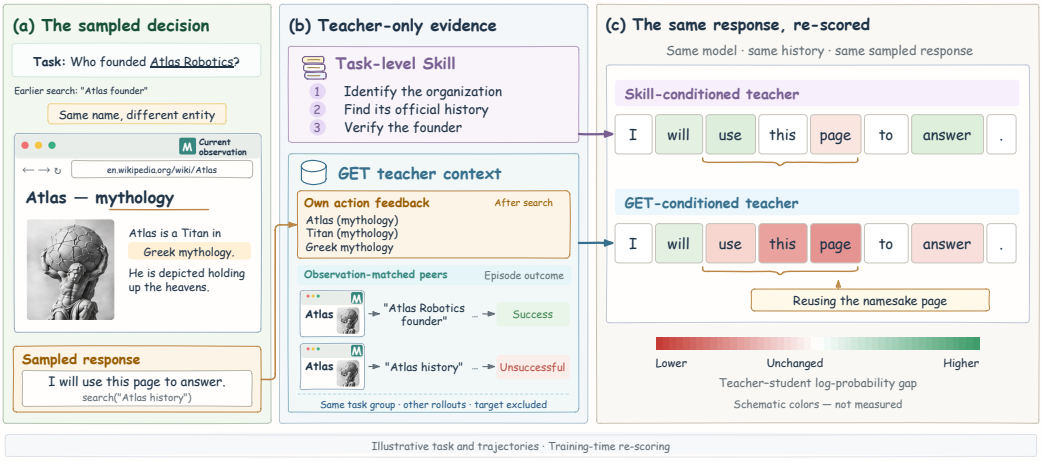


Figure 1: **From task-level knowledge to decision-local evidence.** In this illustrative search interaction, the agent plans to answer from a namesake page and issues an ambiguous follow-up query. A task skill supplies a general procedure; GET combines the query's returned titles with observation-matched peer queries and their source episodes' outcomes. The teacher uses this context to re-score the sampled response. Background colors schematically depict changes in token log probability for its reasoning.

Executing the student's action reveals its immediate consequence, while other attempts record peer actions and their source episodes' outcomes. Matching the current observation brings these peer records together as a reference alongside the student's own feedback (Feng et al., 2025).

We introduce Group Evidence Teacher (GET), which constructs this decision-local reference directly from collected experience. For each response, GET pairs the action's immediate feedback with an Evidence Card drawn from observation-matched transitions in other rollouts of the same task. The card records peer actions and their source episodes' outcomes. Matching and deterministic serialization produce the context without a separate guidance-generation stage. Figure 1 illustrates the role of the peer reference.

The teacher then evaluates the response the student actually sampled. The teacher uses a frozen snapshot of the policy with the richer context, and its scores are compared with the student's ordinary-context token scores. The resulting token log-probability differences determine the weights of an auxiliary student update alongside GRPO's group-relative outcome objective. The learned policy continues to act from its ordinary interaction history.

We evaluate GET on household interaction and search-augmented question answering. On ALFWorld, gated Qwen2.5-3B reaches $77.25 \pm 6.62\%$ success across three training seeds. A matched component comparison shows a 7.66-percentage-point gain from adding peer evidence to action feedback. On Search-QA, GET improves both micro- and macro-averaged exact match over Skill in all four backbone–operator comparisons, including a micro-averaged gain from 40.74% to 42.36% with gated Qwen3-1.7B.

Our contributions are:

- **Direct evidence construction for agentic self-distillation.** We introduce GET, which combines immediate action feedback with observation-matched peer Evidence Cards to inform the same policy's assessment of its sampled responses.
- **Empirical gains in agent learning.** Experiments establish the added value of peer evidence beyond action feedback on ALFWorld and show higher Search-QA accuracy than Skill across both evaluated backbones and weighting rules.
- **Environment-verified task progress and interaction efficiency.** Environment-verified replay of the seed-0 component comparison shows a 13.29-percentage-point gain in required object-state

changes across 143 ALFWorld tasks. On Search-QA, GET with Qwen3-1.7B and Raw OPSD achieves higher test accuracy with 23.0% fewer search calls than Skill.

2 PRELIMINARIES

Multi-turn agents. At turn t of task x, an agent with parameters θ receives observation o_t. Its ordinary context h_t contains the task, current observation, and retained interaction history. The agent samples a response $y_t \sim \pi_\theta(\cdot \mid h_t)$, then parses and executes action a_t from that response. A completed trajectory τ_i records $\{(h_{i,t}, o_{i,t}, y_{i,t}, a_{i,t})\}_{t=0}^{T_i-1}$ and episode return R_i, where T_i is its number of interaction turns.

Group-relative policy learning. To learn from the relative outcomes of multiple attempts, a rollout policy $\pi_{\theta_{\text{old}}}$ samples K trajectories for each task, forming $\mathcal{G}_x = \{\tau_1, \ldots, \tau_K\}$. GRPO uses their returns to assign a trajectory-level advantage to sampled tokens (Shao et al., 2024). With $\bar{R}_x$ and s_x denoting the mean and standard deviation of the group returns, this advantage is $A_i = (R_i - \bar{R}_x)/(s_x + \epsilon)$, where ϵ is a numerical stabilizer. We denote the adopted clipped policy loss (Schulman et al., 2017) and reference regularization by $\mathcal{L}_{\text{RL}}$.

Privileged self-teaching. On-policy distillation obtains teacher supervision on responses generated by the student (Agarwal et al., 2024). Privileged self-teaching supplies additional training-time information when evaluating those responses (Zhao et al., 2026; Hübotter et al., 2026). In the same-policy setting considered here, the teacher view receives ordinary context h and privileged context Z, while the student uses h. The teacher's scores provide supervision on the student's sampled behavior.

3 GROUP EVIDENCE TEACHER

Immediate feedback reveals what happened after the student's sampled action. Other rollouts provide experience of acting in related situations, including peer choices and their source episodes' outcomes. GET brings these two views together so the self-teacher can assess the original response using both its observed consequence and a reference from other attempts.

For decision (i, t), executing action $a_{i,t}$ produces environment feedback $f_{i,t}$. We retain them in a text block $F_{i,t} = \text{Render}(a_{i,t}, f_{i,t})$ and denote the peer Evidence Card by $E_{i,t}$. The privileged teacher context is

$$Z_{i,t} = \text{Concat}(F_{i,t}, E_{i,t}). \tag{1}$$

After collecting the rollout group, the teacher reads $Z_{i,t}$ alongside the original history $h_{i,t}$ to re-score response $y_{i,t}$, which was sampled before the action's execution. If no usable peer card is available, the feedback block remains. Figure 2 connects this context construction to learning.

3.1 SELECTING OBSERVATION-MATCHED PEER EXPERIENCE

Attempts at the same task can reach different situations at the same turn, while matching observations may occur at different turns. A peer reference therefore needs to follow the observation reached by the target decision. Let $c_x(o)$ assign observations to groups within task x. For target (i, t), GET selects the peer visits

$$\mathcal{P}_{i,t} = \{(j, u) : 1 \le j \le K,\ j \ne i,\ 0 \le u < T_j,\ c_x(o_{j,u}) = c_x(o_{i,t})\}. \tag{2}$$

The condition $j \ne i$ excludes the entire target trajectory, while u can differ from t. Groups are rebuilt from each completed rollout batch. The observation locates the records; the target history $h_{i,t}$ provides context for interpreting them. We instantiate c_x with exact serialized observations or thresholded text similarity, as detailed in Appendix A.

3.2 CONSTRUCTING EVIDENCE CARDS

Matched visits may repeat actions and associate them with different episode outcomes. To give the teacher a compact account of these associations, GET deterministically renders the selected

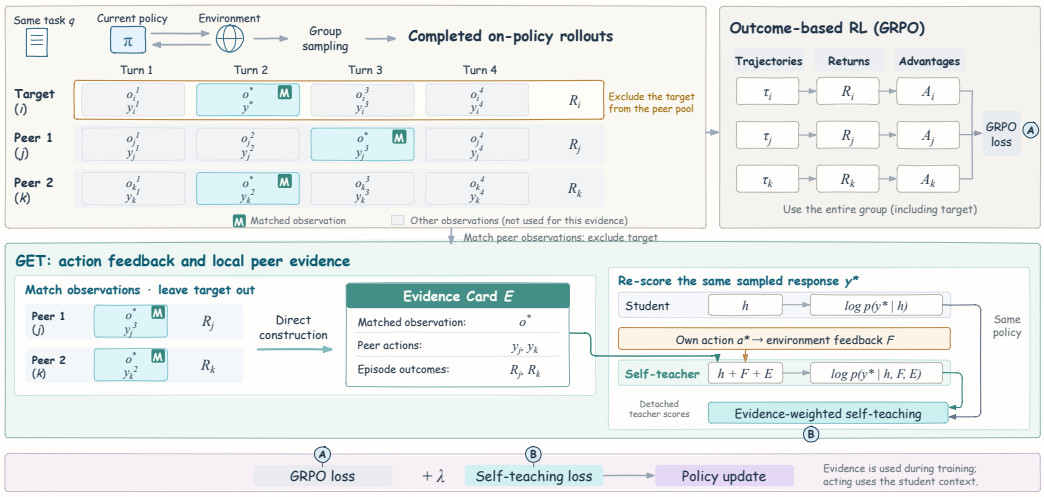


Figure 2: **From collected experience to decision-local supervision.** The rollout group supplies the target action's feedback F and observation-matched transitions from other trajectories, rendered as E. The student scores the original response from ordinary context h; the same policy as teacher additionally conditions on F and E. Token log-probability differences determine auxiliary weights for learning alongside GRPO.

actions, source-episode returns, and support information as an Evidence Card $E_{i,t}$. Trajectory and turn metadata support aggregation and exemplar selection.

For structured actions, repeated choices can be summarized by action. The card reports each canonical action's matched-visit count, the fraction of visits associated with positive episode return, and their mean source-episode return. Genuine repeated visits within a peer trajectory contribute separately to these statistics.

For free-form actions, a concrete peer decision preserves the content of an attempted alternative. The renderer selects an exemplar from a successful peer by discounted episode return and reports its content and support in the matched pool. Without a successful exemplar, it lists observed exploration alternatives. Appendix A specifies action parsing, aggregation, ranking, card availability, and serialization.

3.3 Evidence-Conditioned Self-Distillation

To turn this reference into supervision on the student's own behavior, we evaluate the response it actually sampled. Let k index a response token and $y_{<k}$ denote its sampled prefix. Omitting decision indices, the ordinary and privileged views assign token log probabilities

$$\ell_k^S = \log \pi_\theta(y_k \mid h, y_{<k}), \qquad \ell_k^T = \log \pi_{\bar{\theta}}(y_k \mid h, Z, y_{<k}). \tag{3}$$

Here $\bar{\theta}$ is the actor snapshot after rollout collection and before updates on that batch. Both views use teacher forcing on the original reasoning and action tokens. Teacher scores are computed with gradients disabled and cached; student scores are recomputed in each actor forward pass.

The two scores give a token-specific comparison between the student's ordinary view and the evidence-conditioned teacher. With sg denoting stop-gradient, which preserves a value while removing its derivative, we compute the detached gap

$$\delta_k = \mathrm{sg}[\ell_k^T] - \mathrm{sg}[\ell_k^S]. \tag{4}$$

A positive gap means that the teacher assigns the sampled token higher probability than the student; a negative gap means it assigns lower probability.

To pass this comparison into the update, we map the gap to a token weight w. Raw OPSD uses the signed gap directly; Gate uses SDAR's sigmoid mapping (Lu et al., 2026). With logistic function σ

and positive slope β, the two rules are

$$w_{\text{raw}}(\delta) = \text{sg}[\delta], \qquad w_{\text{gate}}(\delta) = \text{sg}[\sigma(\beta\delta)]. \tag{5}$$

Raw weights can reinforce or suppress sampled tokens. Gate assigns nonnegative imitation weights, so a lower teacher score reduces reinforcement rather than reversing its direction.

These weighted scores supply an auxiliary signal alongside GRPO's trajectory-level objective. Let $m_{i,t} = \mathbb{I}[Z_{i,t} \text{ is usable}]$ indicate availability of the action feedback or peer evidence, broadcast as m_k to response tokens, and let Agg average over all valid response tokens. With auxiliary coefficient λ, we optimize

$$\mathcal{L} = \mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\text{aux}}, \qquad \mathcal{L}_{\text{aux}} = -\,\text{Agg}_k\left[m_k\, w(\delta_k) \ell_k^S\right]. \tag{6}$$

For $m_k = 0$, the auxiliary summand is zero without computing a teacher score or gap; these tokens remain in the averaging denominator. The mask and weights are detached during differentiation, so the constructed context affects the auxiliary gradient through the token weights, while gradients pass through the student scores. At deployment, the learned policy acts from its ordinary interaction history.

4 Experiments

Our experiments ask three questions: **RQ1**: How does GET affect agent performance across tasks and models? **RQ2**: What does peer experience contribute, and how should it be selected? **RQ3**: How does GET affect intermediate task progress and action evaluation? We then examine the learned search behavior and the computational cost of constructing and scoring the teacher context.

4.1 Experimental Setup

ALFWorld and models. We evaluate GET on ALFWorld and search-augmented question answering using Qwen2.5-3B-Instruct (Qwen et al., 2024) and Qwen3-1.7B (Yang et al., 2025). ALFWorld (Shridhar et al., 2021) provides text-based household interaction across six task types: pick-and-place, examination under light, cleaning, heating, cooling, and placing two objects. We train on the official training partition and evaluate all 140 seen and 134 unseen tasks in the official held-out splits. We report task completion within 50 interaction turns, both by task type and over all 274 tasks.

Search-QA. We follow the Search-R1 benchmark (Jin et al., 2025) and retrieval setting used by Feng et al. (2025). Training uses the released NQ and HotpotQA mixture; evaluation covers 51,713 questions from NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle. All policies query the same offline 2018 Wikipedia corpus through an E5-base-v2 retriever (Wang et al., 2022), which returns three passages per search. Agents alternate search and answer actions for up to four turns. We report normalized final-answer exact match per benchmark and over all evaluation questions.

Teaching contexts. GET combines immediate action feedback F and observation-matched peer evidence E. ALFWorld uses the returned observation as feedback; Search-QA uses simple feedback consisting of search-result titles. **Feedback-only** and **Peer-only** retain F and E, respectively. **Skill** uses the same SkillRL-derived SkillBank (Xia et al., 2026) and task-relevant loading procedure released by SDAR (Lu et al., 2026).

Baselines and weighting rules. **GRPO** provides the outcome-supervised baseline. Skill and GET supply privileged context through the same-policy distillation interface in Section 3. **Raw OPSD** uses the signed teacher–student gap directly; **Gate** uses SDAR's sigmoid mapping (Equation 5). Skill with Gate is our SDAR reproduction, labeled **Skill (SDAR)** in the tables. The component study compares Feedback-only with GET; the selector study compares observation-matched and turn-matched peer evidence.

Training and reporting. For the benchmark comparisons, training runs for 150 optimizer steps with eight rollouts per task. ALFWorld and Search-QA use task batches of 16 and 128, respectively, a learning rate of 10^{-6}, and an auxiliary coefficient of 0.01. We evaluate the step-150 policies. For

gated Qwen2.5-3B on ALFWorld, we report three GET and three Skill seeds alongside three GRPO seeds; the Feedback-only component comparison uses seed 0. Update and decoding configurations are given in Appendix B.

4.2 RQ1: Agent Performance across Tasks and Models

Table 1: **Agent performance.** For our evaluations, ALFWorld reports task success over 274 tasks, and Search-QA reports normalized exact match over 51,713 questions. Each configuration uses one fixed step-150 checkpoint from a single training run for all seven benchmarks. All columns use percentages. ALFWorld All and Search-QA Micro weight individual evaluation items equally; Search-QA Macro averages the seven source scores equally. Rows marked † reproduce results from Lu et al. (2026) under the original evaluation protocol; $*$ denotes access to skills at inference. For these rows, ALFWorld All and Search-QA Macro retain the source paper's reported aggregate scores; unreported Search-QA Micro scores are shown as "—". Bold marks the highest value among our evaluations within each backbone. GET and Skill ALFWorld entries use seed 0.

		ALFWorld							Search-QA								
Method	Weighting	Pick	Look	Clean	Heat	Cool	Pick2	All	NQ	Trivia	Pop	Hotpot	2Wiki	MuSiQue	Bam	Micro	Macro
Qwen2.5-3B-Instruct																	
Vanilla†	–	44.4	11.1	6.2	15.4	28.6	12.5	21.9	24.6	48.1	31.0	26.3	25.3	7.2	59.7	–	31.7
Skill-Prompt†,*	–	51.7	66.7	48.4	0.0	4.3	10.0	28.9	23.7	46.2	30.6	24.4	22.1	7.5	12.5	–	23.9
OPSD†	–	48.8	41.7	16.7	0.0	15.8	16.7	28.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	–	0.0
GRPO	–	86.44	74.19	79.31	48.72	**76.09**	65.85	73.36	43.91	59.80	41.83	39.04	38.20	14.15	28.00	43.30	37.85
Skill (SDAR)	Gate	86.44	61.29	82.76	64.10	56.52	39.02	67.52	43.19	58.91	42.20	39.28	37.60	**17.05**	30.40	43.19	38.38
GET	Gate	**89.83**	**83.87**	**89.66**	**82.05**	**76.09**	**80.49**	**84.31**	**45.62**	**61.62**	**45.52**	**39.96**	37.71	14.40	**32.00**	**44.87**	**39.55**
Skill	Raw	79.66	61.29	68.97	69.23	58.70	46.34	65.33	41.30	57.64	41.00	38.38	37.38	15.18	**32.00**	42.18	37.55
GET	Raw	86.44	41.94	72.41	74.36	73.91	56.10	70.07	44.43	59.45	44.06	39.65	**38.52**	14.44	29.60	44.05	38.59
Qwen3-1.7B																	
Vanilla†	–	25.0	22.2	3.1	0.0	21.4	4.2	12.5	29.4	46.9	37.0	23.5	19.6	6.4	10.5	–	24.8
Skill-Prompt†,*	–	10.3	50.0	16.1	0.0	0.0	5.0	9.4	29.4	46.5	36.2	22.9	20.8	4.3	10.1	–	24.3
OPSD†	–	26.3	33.3	9.1	0.0	4.5	5.3	14.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	–	5.8
GRPO	–	**59.32**	**51.61**	31.03	25.64	23.91	17.07	35.40	38.25	55.13	42.73	30.76	31.77	9.10	22.40	39.13	32.88
Skill (SDAR)	Gate	57.63	29.03	41.38	43.59	32.61	**24.39**	39.78	40.30	58.11	42.51	35.65	32.11	10.84	26.40	40.74	35.13
GET	Gate	52.54	38.71	41.38	41.03	45.65	19.51	40.88	**42.91**	**58.83**	**45.27**	**37.12**	**33.07**	12.33	30.40	**42.36**	**37.13**
Skill	Raw	38.98	29.03	56.90	58.97	56.52	4.88	42.34	32.60	48.63	35.82	29.47	30.49	9.93	25.60	34.96	30.36
GET	Raw	40.68	32.26	**60.34**	**61.54**	**69.57**	4.88	**46.35**	39.36	58.80	42.20	36.02	30.96	**12.91**	**33.60**	40.63	36.26

On ALFWorld, GET improves over Skill with both backbones under both weighting rules (Table 1). With Qwen3-1.7B, success increases from 39.78% to 40.88% under Gate and from 42.34% to 46.35% under Raw, gains of 1.10 and 4.01 percentage points, respectively. With Qwen2.5-3B and Gate, the full method achieves $77.25 \pm 6.62\%$ success across three training seeds, compared with $68.37 \pm 1.47\%$ for Skill and $70.07 \pm 3.70\%$ for GRPO (Table 2).

Table 2: **Performance across training seeds.** Qwen2.5-3B on all 274 ALFWorld tasks, using step 150 for every run. GET and Skill use Gate. SD is the sample standard deviation across training seeds. Training and evaluation settings are given in Appendix B.

Method	Seed 0	Seed 1	Seed 2	Mean $\pm$ SD (%)
GET	84.31	76.28	71.17	77.25 ± 6.62
Skill (SDAR)	67.52	67.52	70.07	68.37 ± 1.47
GRPO	73.36	70.80	66.06	70.07 ± 3.70

On Search-QA, GET achieves higher micro- and macro-averaged exact match than Skill in all four backbone–weighting comparisons (Table 1). For Qwen2.5-3B, GET-minus-Skill gains are 1.68/1.17 percentage points (micro/macro) under Gate and 1.88/1.04 under Raw. For Qwen3-1.7B, the corresponding gains are 1.62/2.00 under Gate and 5.67/5.90 under Raw. GET with Gate ranks first under both averages for both backbones.

4.3 RQ2: Peer Information and Selection

Table 3: **Evidence components and peer selection.** Qwen2.5-3B with Gate, seed 0, step 150, full-274 ALFWorld evaluation. A compares GET with each evidence component removed. B compares two peer selectors without action feedback; each selector constructs its own usable cards. Evaluation details are given in Appendix B.

Configuration	Feedback F	Peer evidence E	Success (%)
A: Evidence components			
GET	✓	✓	**84.31**
– Peer evidence	✓	–	76.64
– Action feedback	–	✓	77.74
GET − Feedback-only: +7.66 pp; 95% paired CI [2.55, 12.77]. Task-paired bootstrap, 20,000 resamples; Appendix C.3.			
B: Peer selection without action feedback			
Observation-matched	–	✓	77.74
Turn-matched	–	✓	66.79

Adding peer evidence to action feedback increases task completion from 210 to 231 of 274 tasks (Table 3): a gain of 7.66 percentage points. Gains occur on both seen and unseen tasks, at 7.86 and 7.46 percentage points, respectively (Appendix C.3).

The complete observation-matched configuration achieves 77.74% success, compared with 66.79% for turn matching in the peer-only setting: a difference of 10.95 percentage points. Turn matching takes peer decisions at the same interaction index; observation matching can select visits from different indices. Offline reconstruction on shared archived batches yields lower usable-card coverage for observation matching (Table 5).

4.4 RQ3: Task Progress and Decision Evidence

Where does peer evidence improve task progress? ALFWorld's Clean, Heat, and Cool tasks require the agent to acquire a target object, change its state, and complete its placement. We trace these three milestones in the GET and Feedback-only policies from the component comparison in Table 3: Qwen2.5-3B with Gate, training seed 0, and step 150. Replaying their archived evaluation actions recovers the environment states at each turn. Figure 3 shows the fraction of tasks reaching each milestone across all 143 tasks in these three families.

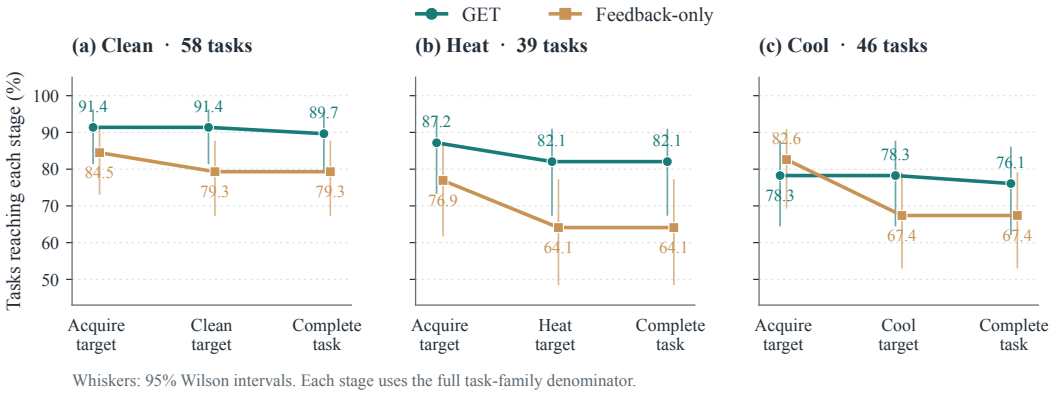


Figure 3: **Task progress with and without peer evidence.** GET and Feedback-only on the Clean, Heat, and Cool tasks from the 274-task ALFWorld evaluation. Points show environment-verified milestone attainment, with all tasks in each family as the denominator; whiskers show 95% Wilson intervals. The pooled GET advantage grows from 4.20 percentage points at acquisition to 13.29 points at the required state change. Appendix C.3 details the milestone measurements.

The advantage expands at the required object manipulation. Across these 143 tasks, GET acquires a target in 123 cases, compared with 117 for Feedback-only, a gap of 4.20 percentage points. GET achieves the required state change in 121 of 143 tasks, compared with 102 for Feedback-only: a gain of 13.29 percentage points (95% task-paired bootstrap CI: $[6.29, 20.98]$; 20,000 resamples). Among trajectories that acquire the target, 121 of 123 (98.4%) complete the manipulation with GET, compared with 102 of 117 (87.2%) with Feedback-only. The learned GET policy more consistently carries acquired objects through the operation required by the goal. GET ultimately completes 119 of these tasks versus 102 for Feedback-only; these three families account for 17 of the 21 additional completions in the full evaluation.

Evidence as evaluation material. An external LLM judge compares GET Evidence Cards with task skills at 750 archived ALFWorld decisions across five interaction stages. It sees the same pre-action context and original response, and chooses which material better supports evaluating that response. GET cards are preferred in 589 comparisons (78.5%), with preferences ranging from 74.7% to 84.0% across stages. Card content further distinguishes the preferences: GET cards are selected in 388 of 403 comparisons with nonzero outcome information (96.3%), versus 201 of 347 with all-zero statistics (57.9%). Appendix C reports sampling, position balancing, and per-stage results.

Task completion after one evidence-conditioned action. At 350 restored ALFWorld states spanning 80 tasks, the same API policy receives a GET Evidence Card or a task skill for its next action, then continues using ordinary observations and recent history. Each state has eight continuations per source, with up to 20 additional actions. GET cards yield 43.82% completion versus 41.71% for Skill, an observed difference of 2.11 points (gamefile-cluster bootstrap 95% interval: $[-0.11, 4.39]$). Table 6 reports the stage-level completion rates.

4.5 Interaction Behavior and Computational Cost

GET combines higher success with fewer search interactions during Qwen3-1.7B training with Raw OPSD (Figure 4). Over the final ten steps, success averages 39.49% versus 37.22% for Skill, while mean search calls fall from 1.696 to 1.332 per trajectory, a 21.5% reduction.

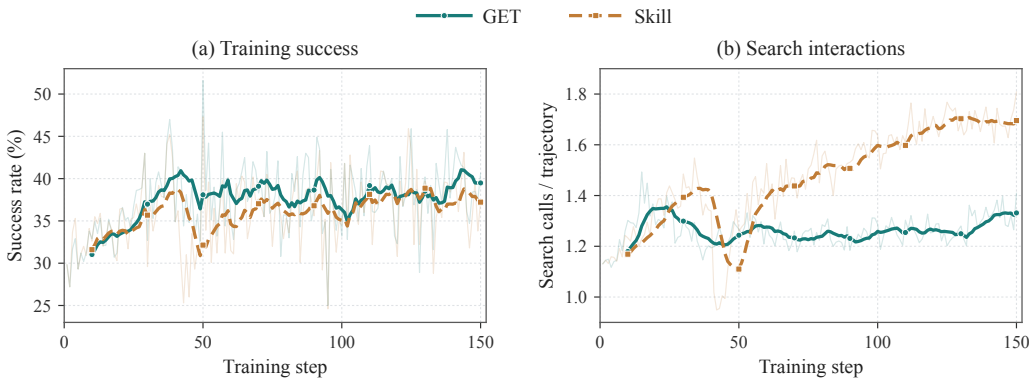


Figure 4: **Higher training success with fewer search calls.** Qwen3-1.7B with Raw OPSD. Both panels use the same 150-step training runs. Faint lines show raw observations; solid and dashed lines show trailing ten-step means. Final-window averages use steps 141–150.

This reduction also appears in the final policy's test behavior. On the 51,713 evaluation questions, GET achieves 40.63% micro-averaged exact match versus 34.96% for Skill, a gain of 5.67 percentage points. Mean search calls across all evaluation questions are 1.390 for GET and 1.805 for Skill, a 23.02% reduction.

Computational cost. We profile a fresh Qwen3-1.7B GET run with Gate on ALFWorld. After five warm-up steps, the next 25 steps average 265.11 seconds each. Evidence-context construction takes 7.35 seconds (2.77%) and teacher scoring takes 19.75 seconds (7.45%); together they account for

10.22% of step time (Figure 5). Matching and rendering use collected records without an additional language-model generation call. Appendix B details the measurement protocol.

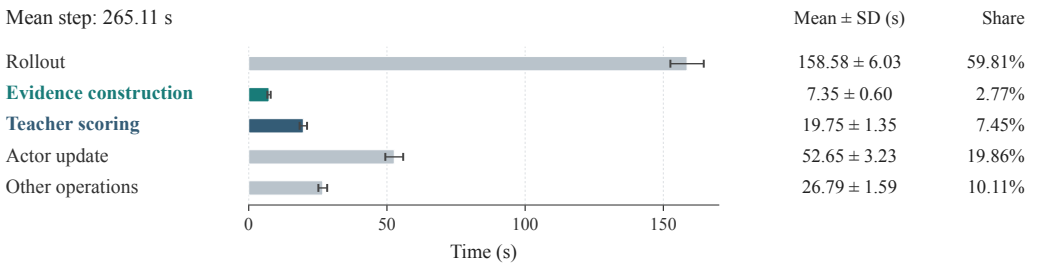


Figure 5: **Time spent constructing and using evidence.** Qwen3-1.7B GET with Gate on ALFWorld. Bars show mean component times; whiskers indicate sample standard deviations across steps 6–30.

5 Related Work

On-policy distillation and feedback. Training on learner-induced trajectories addresses distribution shift in imitation learning (Ross et al., 2011). On-policy distillation supervises student-generated responses (Agarwal et al., 2024); generalized distillation transfers information available only to the teacher (Lopez-Paz et al., 2016). OPSD conditions the policy on privileged context (Zhao et al., 2026), while SDPO supplies environment feedback and successful attempts to the self-teacher (Hübotter et al., 2026). SDAR gates the resulting supervision (Lu et al., 2026); TurnOPD allocates rollout depth and distillation weight across turns (Zhou et al., 2026).

Experience as reusable guidance. Experience can be reused as verbal reflections (Shinn et al., 2023) or hierarchical skills (Xia et al., 2026). Skill-SD retrieves skills for self-distillation (Wang et al., 2026); OPID extracts episode- and step-level hindsight skills from on-policy trajectories (Yang et al., 2026). GET instead renders peer action–episode-outcome records and execution feedback directly.

Peer experience as teaching material. MOPD uses peer successes and failures as distillation context (Yu et al., 2026); SD-Search uses group search traces and outcomes to supervise search-query positions (Ma et al., 2026). GET selects peer visits by the reached observation, across turn positions, and combines their actions and episode outcomes with the target action's feedback.

Local grouping and state-conditioned guidance. Local alignment can serve different roles in learning. GiGPO groups recurring states to derive local scalar advantages (Feng et al., 2025). SMRC-SD matches reached states to compatible positions in successful reference paths and contextualizes their continuations (Liu et al., 2026). GET uses observation groups to select textual peer records for the self-teacher's assessment of the original response. The selected experience changes the auxiliary token weights, while GRPO supplies the outcome advantage.

6 Conclusion

GET turns collected interaction experience into decision-local teacher context. Immediate action feedback records the student's observed execution result, while observation-matched peers provide actions and episode outcomes from related situations. The same policy uses this context to re-score its sampled responses and guide token-level learning. In the matched Qwen2.5-3B ALFWorld seed-0 comparison, peer evidence adds 7.66 percentage points over feedback alone. Search training also combines higher success with 21.5% fewer search calls in the Qwen3-1.7B signed-gap comparison. These results demonstrate the value of directly organizing interaction experience for agent self-distillation.

AI USE STATEMENT

We used generative AI tools for hypothesis refinement, method and experiment design, implementation and debugging, data processing, and interpretation of results. Additional assistance covered literature search, translation, manuscript drafting and revision, and figure preparation. An LLM also served as the external judge in Appendix C. The authors are responsible for verifying all AI-assisted text, code, analyses, and citations, and for the final scientific claims and reported results.

REPRODUCIBILITY STATEMENT

Section 3 specifies GET's evidence construction and learning objective, with task-specific construction details in Appendix A. Appendix B documents training settings, evaluation protocols, and measurement procedures. Appendix C describes decision sampling, judge inputs, environment-verified milestones, and paired bootstrap estimation.

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. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/html/2306.13649v3.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.10978.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. URL https://arxiv.org/html/2601.20802v1.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Ö. Arık, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Conference on Language Modeling*, 2025. URL https://openreview.net/pdf?id=Rwhi91ideu.

Junzhuo Liu, Weiwei Li, Jun Ling, and Peng Wang. When privileged guidance misaligns: State-matched routing and contextualized self-distillation for multi-turn agents. *arXiv preprint arXiv:2608.05219*, 2026. URL https://arxiv.org/abs/2608.05219.

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*, 2026. URL https://arxiv.org/abs/2605.15155.

Yufei Ma, Zihan Liang, Ben Chen, Zhipeng Qian, Huangyu Dai, Lingtao Mao, Xuxin Zhang, Chenyi Lei, and Wenwu Ou. SD-Search: On-policy hindsight self-distillation for search-augmented reasoning. *arXiv preprint arXiv:2605.18299*, 2026. URL https://arxiv.org/abs/2605.18299.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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.

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. URL https://doi.org/10.52202/075280-0377.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

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*, 2026. URL https://arxiv.org/abs/2604.10674.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. URL https://arxiv.org/abs/2504.20073.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026. URL https://arxiv.org/abs/2606.26790.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, and Yu Hu. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026. URL https://arxiv.org/abs/2605.12652.

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.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. TurnOPD: Making on-policy distillation turn-aware for efficient long-horizon agent training. *arXiv preprint arXiv:2607.05804*, 2026. URL https://arxiv.org/abs/2607.05804.

A Evidence Construction Details

A.1 Observation Matching and Action Summaries in ALFWorld

The constructor groups transitions by task identifier and their serialized anchor observation. Actions are parsed from the first action field of the sampled response, lower-cased, and whitespace-normalized. The entire target trajectory is removed before action statistics are computed. Source-row identifiers deduplicate padding copies. Each genuine matched visit contributes one record, including repeated visits within a peer trajectory.

For $\mathcal{P}_{i,t}(a) = \{(j,u) \in \mathcal{P}_{i,t} : a_{j,u} = a\}$, the action summary is

$$n_a = |\mathcal{P}_{i,t}(a)|, \qquad \widehat{s}_a = \frac{1}{n_a} \sum_{(j,u)\in\mathcal{P}_{i,t}(a)} \mathbb{I}[R_j > 0], \qquad \overline{R}_a = \frac{1}{n_a} \sum_{(j,u)\in\mathcal{P}_{i,t}(a)} R_j. \tag{7}$$

Here n_a counts matched visits, $\widehat{s}_a$ is their positive episode-outcome fraction, and $\overline{R}_a$ is their mean episode return. The constructor emits a card when the pool contains at least two distinct nonempty actions. Actions appear in lexical order, with one block per action:

```
<group_evidence>
Cross-fitted statistics from other on-policy trajectories at
this state:
Action: [observed action]
Attempts: [count]
Success rate: [positive-outcome fraction]
Mean return: [mean episode return]
...
</group_evidence>
```

A.2 Observation Matching and Peer Decisions in Search

Within each question, the constructor processes anchor observations in batch order. An observation joins the first existing representative with a Python `SequenceMatcher` ratio of at least 0.90, or becomes a new representative. After grouping, the target trajectory is excluded. The card records the numbers of matched peer trajectories, matched transitions, and successful peer trajectories.

For a pool containing a parseable decision from a successful peer, the selected exemplar maximizes discounted episode return:

$$(j^*, u^*) \in \arg \max_{(j,u)\in\mathcal{P}_{i,t},\ j \text{ successful}} R_j \gamma^{T_j - 1 - u}, \qquad \gamma = 0.95. \tag{8}$$

A search decision contributes its query; an answer decision contributes the decision to answer, with the peer's answer text omitted. The card also states the discounted return and the exemplar's observed support.

When a successful exemplar is unavailable, the card lists up to three distinct peer search queries with their unique-trajectory support counts. A pool without a usable exemplar or query yields an empty card.

A.3 Target Action Feedback

The feedback block precedes the peer card and retains the executed action and its immediate feedback. ALFWorld uses the returned observation and retains the original action text if parsing fails; Search-QA uses simple feedback consisting of search-result titles.

B Training and Evaluation Configuration

Table 4: Training and evaluation settings.

Setting	ALFWorld	Search
Optimizer steps	150	150
Rollouts per task	8	8
Training task batch size	16	128
Learning rate	10^{-6}	10^{-6}
Student prompt / response token limits	2048 / 512	4096 / 512
Maximum interaction turns	50	4
Distillation coefficient λ	0.01	0.01
Gated operator parameter β	5	5
Reference KL coefficient	0.01	0.001
Evaluation decoding	Sampled, $T = 0.4$	Greedy
Evaluation tasks / questions	274	51,713

Skill context. The Skill baseline uses SDAR's released bank and loader with `skill_all=false`, which selects the relevant skill content for the task. The files correspond to upstream revision `06f78586fdeb8f354f4d2339a325640d6668c084`.

ALFWorld protocol. We enumerate all 140 seen and 134 unseen evaluation tasks, with one environment worker per task. Each split is evaluated separately using four GPUs and tensor parallelism of two. The environment seed is zero and worker seeds are $1000 +$ slot. Each task has one sampled rollout with temperature 0.4. Success is the environment's terminal `info.won` value. The combined score divides the total successes by 274. Qwen3 uses `enable_thinking=False`.

Search protocol. The student alternates search and answer decisions for up to four turns. Evaluation uses greedy decoding. Predictions are joined to the canonical question and source identifiers, and the final `<answer>` field is scored with normalized exact match against the canonical gold answer. Each of the 51,713 questions contributes once to the micro-averaged score. Source-level scores use the same predictions, grouped by benchmark. Macro-averaged exact match weights the seven benchmark scores equally.

Three-seed gated repetitions and component comparisons. The full ALFWorld method uses Qwen2.5-3B, seeds 0, 1, and 2, and the step-150 checkpoint for every run. The PPO minibatch size is 256; other settings follow Table 4. The data and environment seeds are 1 and 0. The engine and Python hash seeds vary together; request-local sampling seeds are unset. Feedback-only uses the full seed-0 configuration with the peer card removed.

Training-curve measurements. The Qwen3-1.7B Raw OPSD curves use the original GET and Skill trainer logs, with one recorded point at each optimizer step from 1 through 150. Training success is the rollout metric `episode/success_rate`; search use is `episode/tool_call_count/mean`. The latter accumulates the environment's search-call indicator and averages over unique trajectory IDs, including unsuccessful attempts. It excludes the terminal answer action. Smoothing is a trailing arithmetic mean over ten steps, beginning at step 10; raw points remain visible. Final-window statistics use the unsmoothed points from steps 141–150. Test exact match comes from the separate canonical 51,713-question evaluation at step 150.

Timing protocol. Figure 5 profiles 30 consecutive steps of a fresh Qwen3-1.7B GET Gate run on ALFWorld using four NVIDIA H100 80GB GPUs, tensor parallelism 2, and FSDP actor updates. Settings follow Table 4, with PPO minibatch 256, per-GPU update microbatch 16, and training temperature 1. Validation, checkpoint saving, trajectory archival, and external metric logging are disabled.

Means and sample standard deviations use steps 6–30, after five warm-up steps. Timers measure wall-clock duration, with context construction and synchronous teacher scoring measured separately.

Other operations are the residual after subtracting rollout, construction, scoring, and actor-update times from total step time.

B.1 Offline Selector Coverage

Table 5 compares usable-card coverage for both selectors on shared archived rollout groups from a seed-1 run of the observation-matched peer-only configuration in Table 3B, at updates 1, 75, and 150. Each archived batch contains two complete task groups, with eight rollouts per group. Usable-card coverage is the fraction of recorded decisions for which the constructor emits a nonempty card. Both constructors exclude the target trajectory and require at least two distinct peer actions.

Table 5: **Offline usable-card coverage on shared archived rollout batches.** Percentages refer to decisions with usable cards in the shared archived groups.

Update	Decisions	Observation	Turn
1	800	86.88%	100.00%
75	529	80.15%	94.71%
150	359	60.72%	88.86%

C Decision-Level Evidence Studies

C.1 Archived Experience and Sampling

The judge and continuation studies sample archived Raw OPSD rollouts from Qwen2.5-3B and Qwen3-1.7B. For each backbone, we sample four iterations from each of five training-progress intervals, using the two task groups archived per iteration. Eligible decisions have a nonempty GET card. We sample one decision per trajectory and interaction band, then allocate samples by randomized round-robin over task gamefiles, independently of outcomes and judge preference. Turns are one-indexed and include invalid actions.

The judge study contains 750 decision points from 429 trajectory occurrences and 80 gamefiles. Each band contains 75 decisions from each backbone. The continuation study samples 350 globally distinct trajectories from the 2,422-candidate pool, with 35 states per backbone per band and 80 gamefiles overall. Its state list is frozen before API continuations are collected.

C.2 Judge Inputs and Preference Measurement

The judge receives the student-visible context (normally the last two turns), the original response before its action is executed, and two materials labeled PI A and PI B: the archived GET Evidence Card and the official general-plus-task Skill. The instruction, translated from Chinese, is:

> You are in the teacher position for agent self-distillation and need to evaluate the current action at the given trajectory position. Compare which of the two PI materials better helps judge the quality of the current action. Use the question, history, current action, and PI A and PI B. Return JSON with a preference of A, B, tie, or unknown, and a brief reason.

The API model identifier is `deepseek-flash`; temperature is zero, thinking is disabled, and the output budget is 1,600 tokens. The 750 primary responses all select one of the two materials. GET occupies each A/B position 75 times per band; its preference rates are 73.3% when first and 83.7% when second. Swapping positions for 20 decisions per band preserves the preference in 88 of 100 re-evaluations.

Table 6: **Local evidence across interaction stages.** Left: external-judge preference for GET Evidence Cards over task skills (150 decisions per band). Right: task completion after a single PI-conditioned action by an API policy on a separately sampled cohort (70 states per band, eight continuations per state and source). All rates are percentages; Δ is GET card minus Skill in percentage points. Turns count interactions in the original trajectory.

	Judge comparison		API continuation		
Turn	GET card preferred	Preference rate	GET card	Skill	Δ
1–10	112/150	74.7	53.21	50.71	+2.50
11–20	126/150	84.0	50.18	48.57	+1.61
21–30	116/150	77.3	46.61	47.86	−1.25
31–40	117/150	78.0	30.18	26.43	+3.75
41–50	118/150	78.7	38.93	35.00	+3.93
All	589/750	78.5	43.82	41.71	+2.11

C.3 Environment-Verified Task Progress

We replay the 274-task GET and Feedback-only evaluations used in Figure 3. All 548 replays reproduce the archived initial observations and final success outcomes.

The 58 Clean, 39 Heat, and 46 Cool tasks share three milestones: acquiring a target object, attaining its required state, and completing the task. Acquisition is recorded when the environment's `holds` predicate identifies a task-relevant object. The second milestone requires `isclean`, `ishot`, or `iscool`, respectively, for a target object acquired earlier in the trajectory. Completion uses the environment's `won` signal. We record whether each milestone is reached within the 50-turn evaluation horizon and divide its count by all tasks in the corresponding family. Figure 3 reports these rates with 95% Wilson intervals.

We compute task-paired bootstrap intervals with 20,000 resamples, sampling task IDs with replacement and retaining both policies' outcomes for each sampled task. Table 7 summarizes completion and state-change outcomes.

Table 7: **Paired task outcomes with and without peer evidence.** Qwen2.5-3B, Gate, seed 0, step 150. GET and Feedback-only columns count tasks reaching the indicated milestone; Δ and confidence intervals are in percentage points. Intervals use 20,000 task-paired bootstrap resamples and quantify task-level uncertainty for these fixed policies. "—" denotes an interval not reported.

Task subset	Milestone	Tasks	GET	Feedback-only	Δ	95% CI
All	Completion	274	231	210	+7.66	[2.55, 12.77]
Seen	Completion	140	119	108	+7.86	—
Unseen	Completion	134	112	102	+7.46	—
Clean, Heat, Cool	State change	143	121	102	+13.29	[6.29, 20.98]
Clean, Heat, Cool	Completion	143	119	102	+11.89	[4.90, 18.88]