GRAPHOPD: GRAPH-AUGMENTED ON-POLICY DISTILLATION FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation post-trains large language model agents by supplying dense, step-level guidance from a teacher policy when the reinforcement-learning reward is sparse and arrives only once per trajectory. Existing instantiations allocate this guidance by the size of the teacher-student divergence at each step, on the single-turn intuition that a large disagreement marks a mistake worth correcting. Once decisions chain over many turns, that rule misfires, since an early drift enters every later context both policies condition on, leaving the teacher consistent with the drifted trajectory instead of flagging its cause, while interchangeable steps register large but outcome-irrelevant divergences. We demonstrate this on an agentic benchmark, where distilling the highest-divergence steps brings no consistent benefit over random selection. To this end, we introduce **GraphOPD**, the first method to bring graph-based structural augmentation into on-policy distillation for agent capabilities. It reads which steps enabled which later ones from the environment's own record of state changes, immune to the drift that corrupts the teacher-student gap, organizes them into a dependency graph, scores each step by a random-walk stationary distribution over it, and fuses that structural credit with the divergence signal into a trajectory-relative mask concentrating supervision on each rollout's highest-aptitude steps. Across three model scales and eleven baselines on ALFWorld, WebShop, and SearchQA, GraphOPD shows competitive performance throughout, improving over the strongest baseline by up to **+5.8 pp**. An executed-replay audit further shows that this structural credit score tracks true causal impact far above chance, that both fused signals are independently necessary, and that the same signal transfers to out-of-domain tool-integrated reasoning.The code is at https://anonymous.4open.science/status/GraphOPD-5011.

1 INTRODUCTION

On-policy distillation has become a standard way to post-train large language model agents that pursue goals through extended interaction with an environment (Yao et al., 2023), spanning household task execution, multi-step retrieval, and web purchasing (Shridhar et al., 2021; Yao et al., 2022; Yang et al., 2018). Reinforcement learning provides the outcome-level training signal for such agents (Shao et al., 2024), and because that signal is sparse and arrives only once per trajectory, on-policy distillation complements it with dense, step-level supervision from a teacher policy, typically a frozen snapshot of the student matched via KL divergence at every step (Zhao et al., 2026; Song & Zheng, 2026; Li et al., 2026c). This combination substantially improves sample efficiency on agentic benchmarks (WANG et al., 2026; Liang et al., 2026), yet allocating that step-level supervision across a trajectory remains an open and challenging problem central to the whole recipe.

Existing agentic methods answer it by weighting each step by the magnitude of its teacher-student divergence, on the single-turn intuition that a large disagreement marks a mistake worth correcting (Zhong et al., 2026; Lu et al., 2026; Li et al., 2026a; Zhang et al., 2026b), while a parallel line keeps the same magnitude rule but acts on it indirectly, routing the gap into the RL update or enriching the teacher with privileged skills or debate (Yang et al., 2026a;b; Wang et al., 2026c;d). A separate family reweights the RL advantage itself with structure read from the rollout, such as branching or graph-based credit scores, but leaves the distillation loss unweighted (Wang et al., 2026b; Yuan et al., 2025). Across all of these lines the allocation rule remains the single-turn one, in

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.

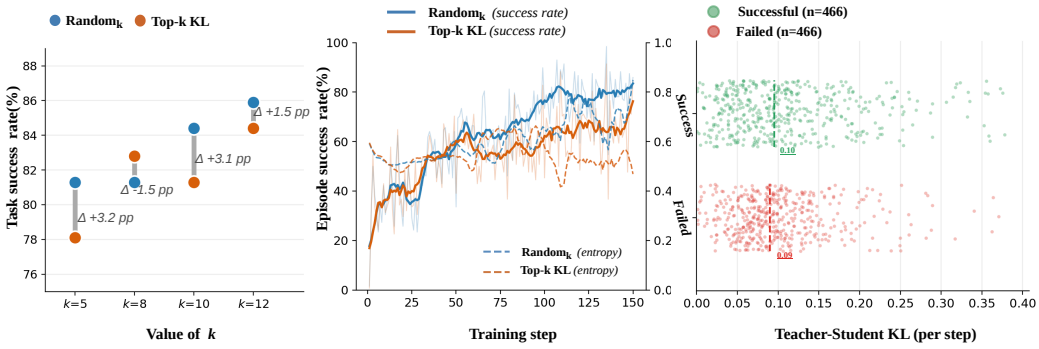


Figure 1: **Motivating empirical study.** All runs distill only the k highest-KL steps or k randomly chosen steps per trajectory, $k \in \{5, 8, 10, 12\}$, on ALFWorld with a 7B Qwen2.5-Instruct agent under GRPO for 150 steps, all other conditions identical. (**Left**) Random selection stays ahead at every budget k, by $+1.5$ to $+3.2$ pp in success rate, with the per-budget gaps Δ annotated above each pair. (**Middle**) At $k = 5$ the KL-guided run trails in success rate (solid curves, left axis) throughout and ends at 76.3% against 83.5% for random, while its policy entropy (dashed curves, right axis) contracts as the random run's expands. (**Right**) The per-step KL of 466 successful and 466 failed steps from 50 rollouts overlaps by 84% between the two outcomes, with medians (dashed) at 0.10 and 0.09, so success and failure are barely separable by divergence.

which the size of the teacher-student gap, alone or combined with rollout structure on the RL side, decides where supervision goes.

As shown in Figure 1, we test that rule where decisions chain over many turns and find that it fails exactly as the structure of agent trajectories predicts , since distilling the k highest-KL steps brings no consistent benefit over distilling k random steps, the divergence-guided rule trails random selection throughout training while policy entropy collapses rather than grows, and successful and failed steps barely separate by divergence. An early drift enters every later context both policies condition on, leaving the teacher locally consistent with the drifted trajectory precisely at the outcome-deciding steps, and interchangeable steps register large divergences unrelated to the result, so divergence magnitude measures how far the student has drifted rather than whether that drift mattered. Agent trajectories nevertheless record what divergence cannot see, namely which steps enabled which later ones, readable from environment state transitions rather than from either policy's predictions, and no prior agentic on-policy distillation method has used this record to place the teacher's supervision.

To this end, in this paper we focus on placing the step-level supervision where the teacher's guidance remains usable along a multi-turn trajectory, and propose **GraphOPD**, the first method to introduce graph-based structural augmentation into on-policy distillation for improving agent capabilities. For each rollout, GraphOPD reads which earlier state changes a later action consumes into a directed dependency graph, converts its random-walk stationary distribution into a per-step structural credit score, and fuses that score with the divergence signal into a trajectory-relative mask that spends supervision where the teacher's guidance is usable, leaving the rest to the RL objective. The structural score marks the steps at which much of the trajectory's accumulated dependency structure is resolved, so it stays informative exactly where the teacher-student gap has been blinded by drift or inflated by interchangeable actions. Since both signals come from the rollout itself, the mechanism needs no extra model calls and enters training as a drop-in reweighting of the distillation loss. We evaluate GraphOPD on ALFWorld, WebShop, and SearchQA across three model scales against eleven baselines, where it shows competitive performance throughout and improves over the strongest baseline by up to **+5.8 pp**. Analyses further confirm that both signals are independently necessary, that the structural credit score tracks true causal impact far above chance under an executed-replay audit, and that the same signal transfers to out-of-domain tool-integrated reasoning.

The main contributions of this paper are summarized as follows:

- We study the divergence-magnitude rule on multi-turn agent benchmarks and find that the teacher-student gap measures drift rather than outcome relevance, so distilling its highest-

divergence steps brings no consistent benefit over random selection, motivating a drift-immune supervision signal for long-horizon agents.

- We are the first to bring graph-based structural augmentation into on-policy distillation for agents, reading a directed graph from environment state transitions to capture which steps the outcome turns on.
- We design GraphOPD, which fuses a random-walk structural credit score with the divergence signal into a distillation aptitude and converts it into a trajectory-relative mask that trains alongside the reinforcement-learning objective.
- Experiments across three benchmarks, three scales, and eleven baselines show gains up to +5.8 pp over the strongest baseline, with the structural credit score validated by executed-replay counterfactuals and out-of-domain transfer.

2 Related Work

On-policy distillation for LLM agents. On-policy distillation augments reinforcement learning with dense token-level supervision from a teacher evaluated on the student's own rollout states (Song & Zheng, 2026; Li et al., 2026c), and the methods we compare against are either instances of it or bounds on what it can add. Objective-level work stabilizes or generalizes the same recipe via target reformulation, entropy-aware divergences, teacher-supported truncation, or reward extrapolation (Jang et al., 2026; Jin et al., 2026; Fu et al., 2026; Yang et al., 2026c). Among the on-policy methods, OPSD (Zhao et al., 2026) distills the student's own frozen snapshot over its rollouts and SDAR (Lu et al., 2026) applies the same recipe to multi-turn agent rollouts, RLSD (Yang et al., 2026a) routes the teacher-student gap into the RL objective as advantage shaping or update magnitude, and OPID (Yang et al., 2026b) distills on-policy from a teacher augmented with privileged skill context. Skill-GRPO and Skill-SD (Wang et al., 2026c) injecting external skill demonstrations or conditioning self-distillation on them. Each on-policy baseline allocates supervision by the magnitude of the teacher-student divergence, which is the shared premise our empirical study tests.

Step-level weighting and structural credit assignment. Beyond our comparison set, on-policy distillation has been extended along three axes, controlling when to distill through temporal curricula (WANG et al., 2026; Liang et al., 2026), how strongly to supervise each step from the divergence through gating, granularity adaptation, or advantage routing (Zhong et al., 2026; Li et al., 2026a; Zhang et al., 2026b), and how reliable the teacher is through debate, rollout-tested guidance, or bounded update magnitudes (Wang et al., 2026d; Chen et al., 2026; Liu et al., 2026; Wang et al., 2026c; Sun et al., 2026). A separate family reweights the RL advantage rather than the distillation loss using structure read from the rollout: branching and graph-based credit scores (Wang et al., 2026b; Yuan et al., 2025; 2026; Li et al., 2026b; da Costa Cunha et al., 2026), dependency and memory graphs (Wang et al., 2026a; Nussbaum-Hoffer et al., 2026; Wang et al., 2026b), verifier- or Shapley-based edge credit (Wang et al., 2021), milestone and rubric judgments (Zhang et al., 2026a), and bisimulation or transport distances to a reference plan (Wang et al., 2026a). Beyond agent RL, graphs underlie self-correcting knowledge-graph planning (Chen et al., 2024), multi-modal entity alignment (Chen et al., 2020; 2022), temporal or de-correlated recommendation (Zhang et al., 2024; Wu et al., 2024), and video social-graph generation (Wu et al., 2021). None of these represent, for the distillation loss, which steps the later ones actually depend on, and that is what GraphOPD reads directly from the trajectory.

3 Method

GraphOPD fuses two per-step signals into a single distillation weight. The first is a structural credit signal that identifies which steps in a trajectory were critical to its outcome, derived from a lightweight dependency graph built over the steps' entity-level relations and scored with a random-walk stationary distribution. The second is the divergence signal used by prior on-policy distillation methods, measuring how far the student's behavior has drifted from the teacher's. GraphOPD combines the two into a trajectory-relative per-step weight that spends distillation only where the teacher's guidance is both trustworthy and useful, leaving the rest to the RL objective. Figure 2 summarizes the full pipeline.

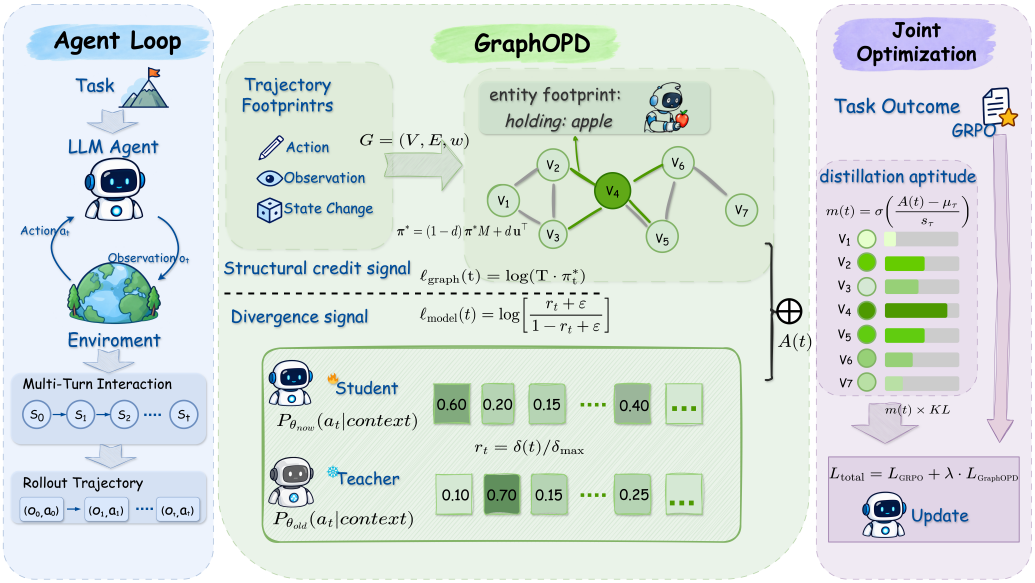


Figure 2: **GraphOPD overview.** From a single rollout, two per-step signals are computed in parallel, a structural credit signal from the Footprint-built dependency graph and its random-walk stationary distribution, and a divergence signal from the teacher-student gap normalized within the trajectory. The two are fused into a distillation aptitude $A(t)$, converted into a trajectory-relative mask $m(t)$, and the resulting masked distillation loss is added to the unweighted GRPO objective; every stage reads only the rollout itself and needs no extra model calls.

3.1 Problem Setup

We consider an agent that interacts with an environment over T steps and produces a trajectory $\tau = (s_0, a_1, o_1, \ldots, a_T, o_T)$, where at each step t, against environment state s_t, the agent emits action a_t and the environment returns observation o_t; the action and observation text are what our Footprint extractor reads. A terminal reward R, issued only once at the end of the episode, encodes task success or failure, and no dense per-step reward is assumed.

The student policy π_θ is trained under GRPO, augmented by a teacher signal at each step. We take the teacher to be $\pi_{\theta_{\text{old}}}$, the student's snapshot frozen at the start of each rollout batch and already used by GRPO for the importance ratio. Staying within one update of the current policy, it serves as an on-support behavioral anchor rather than an external expert.

3.2 Structural Credit Assignment

Assigning credit without a dense reward. With only a terminal reward, a trajectory offers no direct signal for which of its steps actually mattered. Agent actions constrain what is possible later, and this directed structure is readable from action and observation text alone, without LLM calls, privileged annotations, or rollout counterfactuals. We use it as a proxy for step-level credit, made explicit through a lightweight graph $G = (V, E, w)$, with one node $v_t \in V$ per step, built from a rule-based *State Transition Footprint* $\mathcal{F}(t)$ of entity-level state-change keys

$$\mathcal{F}(t) = \mathcal{F}_{\text{env}}(a_t,\, o_t) \;\subseteq\; \big\{\, \texttt{<type>:<entity>} \,\big\}. \tag{1}$$

Full extractor specifications are in Appendix A.

Edge construction. A directed edge $v_i \to v_j$ $(i < j)$ is added whenever step i's state change is explicitly referenced by step j's action,

$$(v_i \to v_j) \in E \iff \exists\, \texttt{<type>:<entity>} \in \mathcal{F}(i) \text{ s.t. } \texttt{<entity>} \in \text{substr}(a_j). \tag{2}$$

Two auxiliary types complete the graph. *Backtrack edges* of weight 0.5 connect j back to an earlier i whenever $o_j = o_i$, marking a revisit, and *sequential edges* of weight 0.05 connect consecutive steps lacking a dependency or backtrack edge, preserving connectivity for the centrality computation.

Soft pruning by entity specificity. Substring matching can admit spurious dependency edges through high-frequency entities that co-occur with many steps without carrying a specific dependency. Rather than a constant weight, we weight each dependency edge by the rarity of the entity that induces it. Let $df(e)$ be the number of steps in the trajectory whose action or observation mentions entity e, and the dependency-edge weight is

$$w(v_i \to v_j) = \frac{1}{df(e)}, \tag{3}$$

where e is the matched entity, or the rarest one if several match. An entity mentioned once or twice yields weight near 1, and a ubiquitous entity yields a small weight that is diluted once the stationary distribution's row normalization is applied. This soft pruning introduces no hyperparameter and leaves the backtrack and sequential edges unchanged.

What the dependency graph encodes. The edges of G record which earlier state changes a later action actually consumes. That is an enabling relation in the environment dynamics and a factual property of the recorded rollout, so it does not inherit the noise affecting the KL-based divergence signal in long-horizon settings. We accordingly treat ℓ_{graph} as a *structural credit score*, measuring how much of the trajectory's later behavior runs through a given step. Whether that structural quantity tracks the steps the outcome really turns on is an empirical question, and we settle it with an executed-replay counterfactual audit in our experiments. Because a step's stationary mass is the rarity-weighted inflow of every dependency chain its action consumes, intermediate milestones at which several upstream state changes meet score above the neutral point just as terminal actions do, and the trajectory-relative mask ranks steps within their own rollout rather than by position.

3.3 Structural Credit and Divergence Signals

Structural credit signal ℓ_{graph}. Counting only direct dependency edges misses transitive importance, so we use a random-walk stationary distribution over the graph, which propagates influence along edges rather than only counting them. Let $M \in \mathbb{R}^{T\times T}$ be the row-normalized edge-weight matrix, in which leaf nodes receive a uniform row $1/T$. With damping $d = 0.15$ and uniform teleport $\mathbf{u} = [1/T, \ldots, 1/T]^\top$, the stationary distribution satisfies

$$\boldsymbol{\pi}^* = (1-d)\,\boldsymbol{\pi}^* M + d\,\mathbf{u}^\top. \tag{4}$$

Mass flows forward along the dependency edges, peaking at convergence points where the preconditions assembled earlier are exercised and the outcome is decided. We convert it to log-odds relative to the uniform baseline,

$$\ell_{\text{graph}}(t) = \log(T \cdot \pi^*_t), \tag{5}$$

which gives $\ell_{\text{graph}}(t) > 0$ above average and 0 at uniform, matching the neutral point of the divergence signal defined next.

Divergence signal ℓ_{model}. The divergence signal measures how much the student's step-t behavior differs from the teacher's. Raw KL is dominated by global training-stage drift, so we use each step's deviation relative to the trajectory, namely the mean absolute token log-probability difference over the student's action tokens S_t,

$$\delta(t) = \frac{1}{|S_t|}\sum_{k\in S_t}\left|\log \pi_{\theta_{\text{old}}}(k \mid s_t) - \log \pi_\theta(k \mid s_t)\right|. \tag{6}$$

We then convert the relative deviation $r_t = \delta(t)/\delta_{\max} \in [0,1]$ to a smoothed within-trajectory log-odds score

$$\ell_{\text{model}}(t) = \log\left[\frac{r_t+\varepsilon}{1-r_t+\varepsilon}\right], \quad \delta_{\max} = \max_{t'\in\tau}\delta(t'), \tag{7}$$

with smoothing floor $\varepsilon = 0.1$. The score is 0 at mid-deviation, where $r_t = \frac{1}{2}$, matching ℓ_{graph}'s neutral point, and the floor bounds it to $\pm\log\frac{1+\varepsilon}{\varepsilon} \approx \pm 2.4$, the same order as $|\ell_{\text{graph}}| \le \log T$. The two signals are therefore commensurate and either source can veto the other, while the within-trajectory normalization keeps them comparable across training stages.

3.4 Distillation Aptitude, Masking, and Training Objective

Fusing the two signals into a distillation aptitude. Both signals already share a common footing, since each is zero at its own neutral point and positive above it, and the ε-floor in Eq. equation 7 keeps ℓ_{model} on the same order as ℓ_{graph}'s $\pm \log T$ range established above, so neither can unilaterally dominate the other. We combine them by a weighted sum

$$A(t) = \alpha \cdot \ell_{\text{model}}(t) + (1 - \alpha) \cdot \ell_{\text{graph}}(t), \tag{8}$$

the *Distillation Aptitude*. $A(t)$ is high when both sources agree, meaning the student has drifted *and* the step is central, and it stays attenuated when the two disagree. α is selected by grid search on a validation split, with $\alpha = 1$ recovering divergence-only weighting and $\alpha = 0$ recovering pure structural-credit weighting.

Trajectory-relative mask and training objective. The absolute scale of $A(t)$ is not stable across training, because δ_{max} and the graph's stationary distribution both shift as the policy changes, so a fixed threshold on $A(t)$ would need constant re-tuning. We instead standardize $A(t)$ within its own trajectory before masking. With μ_τ and s_τ the mean and standard deviation of $\{A(t)\}_{t=1}^{T}$ over τ,

$$m(t) = \sigma\left(\frac{A(t) - \mu_\tau}{s_\tau}\right), \tag{9}$$

so $m(t) > \frac{1}{2}$ exactly when step t's aptitude exceeds its own trajectory's average, and this relative threshold stays meaningful at every stage of training because it is defined against τ itself rather than against a fixed scale. Every trajectory thus spends its own distillation budget on its own most decisive steps, independent of trajectory length, dependency density, or how far training has progressed. Design and numerical details, including the degenerate $s_\tau = 0$ case, are deferred to Appendix B.

The distillation loss aggregates the masked forward KL over each trajectory step, normalized by the total mask weight,

$$\mathcal{L}_{\text{GraphOPD}} = \frac{1}{\sum_t m(t)} \sum_{t=1}^{T} m(t) \cdot \mathrm{KL}\big(\pi_{\theta_{\text{old}}}(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t)\big). \tag{10}$$

This loss is combined with the GRPO policy gradient objective as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GRPO}} + \lambda \cdot \mathcal{L}_{\text{GraphOPD}}, \tag{11}$$

where $\lambda = 0.05$ sets the weight of the distillation term relative to the RL objective and is held fixed across all benchmarks and scales. GraphOPD is implemented as a drop-in wrapper around the training loop, and Algorithm 1 summarizes the full pipeline.

4 Experiments

4.1 Experimental Setup

We evaluate on three agentic benchmarks, ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and SearchQA, a suite of seven open-domain question-answering datasets (Kociský et al., 2018; Joshi et al., 2017; Mallen et al., 2023; Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023), scored by episode success rate, by success rate together with completion score, and by exact-match accuracy respectively; the per-benchmark average columns are the headline numbers quoted in the text. All methods share the same GRPO (Shao et al., 2024) backbone implemented in `veRL` (Sheng et al., 2025) and are trained at three model scales, Qwen3-1.7B (Yang et al., 2025), Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024), against eleven baselines; baseline families, prompts, training configurations, and the run-variance table are detailed in Appendix C.

4.2 Main Results

GraphOPD turns the dependency structure of a trajectory into a training signal that existing on-policy distillation methods do not use, and Table 1 shows this signal translating directly into consistent gains. The advantage is most direct on the two benchmarks that require sustained manipulation of an environment over many turns, where GraphOPD attains the best average success rate

Table 1: **Main results.** Success rate (ALFWorld), accuracy (SearchQA), and success rate (WebShop) across three model scales.**Bold** and underline marks the best and the second best result per column within each model block.

	ALFWorld							SearchQA								WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg	NarQA	Triv	Pop	Hotp	2Wk	MuS	Bam	Avg	Score	Succ.
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	6.7	0.8
GRPO	91.2	62.5	96.2	61.9	65.0	47.4	75.0	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
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	0.2	0.8
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	11.3	3.1
Skill-GRPO	94.3	57.1	**100.0**	66.7	73.1	57.1	80.5	44.3	59.6	44.3	39.0	36.1	14.5	14.9	36.1	76.3	66.4
GRPO+OPSD	**100.0**	82.4	85.7	75.0	70.0	60.0	81.2	44.9	61.2	**45.2**	40.4	38.5	16.0	66.1	44.6	77.8	66.4
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9	73.4	44.4	60.4	44.0	39.5	40.4	15.4	64.9	44.1	75.9	64.0
RLSD	87.9	75.0	90.9	75.0	73.1	68.4	79.7	41.5	58.6	42.3	40.4	40.2	16.8	**66.9**	43.8	84.4	66.4
SDAR	85.2	69.8	88.6	**85.6**	80.4	74.1	81.8	43.7	59.2	43.9	38.0	36.4	14.9	63.7	43.4	83.8	69.3
OPID	94.2	**100.0**	90.0	66.7	74.1	63.1	82.8	**45.1**	61.4	43.4	40.5	38.9	17.2	60.3	44.6	83.7	74.5
GEPO	90.7	76.4	89.5	64.8	78.2	72.0	80.7	43.7	**62.1**	44.4	40.2	37.5	**17.9**	65.0	44.6	88.1	75.3
Ours	90.5	73.5	92.9	**85.6**	**89.6**	**88.3**	**88.6**	43.8	60.3	45.0	**41.0**	**41.8**	17.4	64.6	**46.1**	**90.7**	**78.9**
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	12.5	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
GRPO	91.2	87.5	96.2	81.0	65.0	57.9	81.2	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0	80.9	72.6
Skill-Prompt	51.7	50.0	32.3	5.3	4.3	0.0	23.4	30.9	52.1	32.7	32.7	27.9	12.7	66.1	36.4	1.7	0.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	32.8	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
Skill-GRPO	**100.0**	83.3	96.4	83.3	75.0	78.9	88.3	44.8	63.0	45.1	43.7	43.7	20.5	71.4	47.5	87.0	81.2
GRPO+OPSD	91.4	61.5	**100.0**	87.5	76.5	52.2	80.4	**47.3**	**64.5**	46.9	43.8	39.3	18.0	69.4	47.0	86.8	76.5
Skill-SD	93.9	**93.8**	90.9	**100.0**	69.2	68.4	85.1	47.1	**64.5**	**47.8**	44.2	42.1	20.2	69.0	47.8	86.1	76.5
RLSD	**100.0**	87.5	92.3	58.8	80.0	65.2	82.0	46.8	63.0	44.4	**45.5**	**48.9**	**21.5**	**73.0**	**49.0**	87.4	77.3
SDAR	91.1	78.9	97.2	82.8	75.5	76.8	84.4	46.4	63.0	44.7	43.2	44.5	19.4	64.1	47.5	88.0	82.0
OPID	97.5	80.0	96.8	93.6	86.9	78.5	89.1	47.2	64.2	46.1	44.4	43.0	20.1	67.8	48.2	87.0	80.0
GEPO	90.7	80.1	96.3	77.3	83.4	79.2	85.9	43.9	61.6	44.1	40.8	41.3	17.5	65.2	45.5	87.1	80.0
Ours	97.8	70.6	94.5	97.2	**92.5**	**88.9**	**91.9**	46.4	63.3	45.2	43.2	42.9	19.2	64.0	47.3	**90.4**	**83.6**
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	46.5	4.7
GRPO	71.1	41.7	36.4	40.0	31.8	31.6	46.1	40.0	58.9	43.5	35.4	30.3	12.0	65.7	40.8	67.3	38.3
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	23.0	2.3
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	47.4	9.3
Skill-GRPO	31.4	42.9	51.9	8.3	11.5	7.1	28.1	38.0	58.4	43.9	36.3	29.0	12.5	**66.9**	40.7	80.4	50.0
GRPO+OPSD	38.2	50.0	30.8	28.6	30.0	21.1	32.0	40.7	58.9	45.0	37.0	34.6	13.3	65.7	42.2	70.7	38.3
Skill-SD	52.9	37.5	69.2	42.9	60.0	36.8	52.3	39.1	57.5	**45.4**	34.8	34.1	10.7	64.1	40.8	81.8	53.9
RLSD	50.0	37.5	61.5	19.0	50.0	21.1	42.2	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6	74.0	50.8
SDAR	**77.0**	48.2	61.4	34.1	32.8	31.3	51.3	38.7	57.8	41.8	35.2	33.9	12.9	62.1	40.7	76.7	59.9
OPID	75.2	**59.0**	61.9	62.5	50.9	**37.9**	58.2	37.3	57.6	40.4	34.7	31.5	13.0	62.9	39.6	76.7	66.4
GEPO	72.9	54.0	**71.6**	**65.9**	45.9	35.4	58.1	39.4	59.0	42.0	36.3	34.9	13.9	53.5	40.1	82.6	67.9
Ours	74.5	56.7	71.1	61.1	**70.9**	37.1	**63.3**	**42.8**	**59.6**	40.3	**39.3**	**35.1**	**16.3**	59.3	**42.9**	**85.7**	**73.4**

on ALFWorld and the best success rate on WebShop at every model scale, ahead of all eleven baselines, peaking at 91.9% ALFWorld success and 83.6% WebShop success at 7B. The margin over GEPO, the closest prior baseline, which reweights the GRPO advantage with a graph-based credit score rather than adding a distillation loss, is positive at every scale, reaching +5.2 to +7.9 pp on ALFWorld and +3.6 to +5.5 pp on WebShop success rate. GraphOPD also outperforms the skill-augmented baselines that receive privileged demonstrations or validated skill libraries, such as Skill-GRPO and GRPO+OPSD, on both benchmarks at every scale, despite reading its own signal for free from the rollout rather than from external supervision. The same structural signal also carries over to SearchQA, a retrieval-and-reasoning setting with a substantially different action space, where GraphOPD leads at 3B and 1.7B with 46.1 and 42.9 average accuracy, ahead of every direct-comparison baseline, and stays competitive at 7B. The margin over SDAR is positive on both benchmarks at every scale, at **+12.0 pp** on ALFWorld and **+13.5 pp** on WebShop at 1.7B, **+6.8 pp** and **+9.6 pp** at 3B, and **+7.5 pp** and **+1.6 pp** at 7B. The run-to-run standard deviations behind these means, given in Table 7, confirm that the comparisons sit within reasonable fluctuation: GraphOPD's own spread never exceeds 1.6 pp and stays below 0.6 pp in seven of the nine headline cells, well under half the largest baseline spreads of about 5 pp.

4.3 Ablation Study

We ablate four design choices to isolate each component's contribution, and add a shuffled-centrality control to test whether the gain comes from the structural credit signal's content or from the mere act of non-uniform weighting. Table 2 reports results on WebShop, SearchQA, and ALFWorld. Divergence-only, which drops the structural credit signal, costs 3.9 to 10.5 pp, so the structural credit signal carries substantial information about which steps matter. Structural-only, which drops

Table 2: **Ablation study.** Success rate (WebShop), accuracy (SearchQA), and success rate (ALFWorld). Variant definitions are given in the text above, and the GraphOPD row reproduces the three-run means of the full method, while each ablation variant is a single-run point estimate. **Bold** marks the best result per column within each model block.

	Qwen3-1.7B			Qwen2.5-3B			Qwen2.5-7B		
Method	WebShop	SearchQA	ALFWorld	WebShop	SearchQA	ALFWorld	WebShop	SearchQA	ALFWorld
GraphOPD	**73.4**	**42.9**	**63.3**	**78.9**	**46.1**	**88.6**	**83.6**	**47.3**	**91.9**
Soft-Pruning	71.1	37.4	62.5	77.3	44.2	85.9	82.0	45.3	90.6
Divergence-only	64.8	32.4	59.4	74.2	40.8	84.4	79.6	42.3	87.5
Structural-only	69.5	36.4	60.2	75.0	43.3	84.4	81.3	43.6	89.1
Shuffled Centrality	68.8	40.1	59.4	72.7	41.2	82.8	78.1	41.9	85.9

the divergence signal, is also necessary at 2.3 to 6.5 pp, and the two signals therefore act as complements rather than substitutes. Soft-Pruning, which replaces the entity-specificity dependency-edge weighting with a constant weight, costs 0.8 to 5.5 pp, less than dropping either signal outright at every one of the nine benchmark-scale cells. Shuffled Centrality keeps the mask's sparsity and weight distribution but randomizes which step receives which weight. At 7B, and on WebShop and ALFWorld at 3B, it costs *more* than Divergence-only by up to 1.6 pp, so misassigning the structural credit scores to the wrong steps hurts more than having no structural signal at all. At 1.7B, and on SearchQA at 3B, the pattern reverses and Shuffled costs no more than Divergence-only, most sharply on SearchQA at 1.7B, where the two cost 2.8 pp and 10.5 pp respectively. At the smallest scale, then, the redistribution of supervision weight itself carries part of the structural credit signal's benefit.

Hyperparameter sensitivity. We select the mixing coefficient α by grid search on the ALFWorld validation split, fixing the distillation loss coefficient $\lambda = 0.05$, and the selected configuration of $\alpha = 0.5$ with $\lambda = 0.05$ is reported alongside the α sweep. The sweep confirms that the gain comes from fusing both signals, since on both ALFWorld and WebShop in Table 3 every other α underperforms the selected $\alpha = 0.5$, with a consistent asymmetric cost at the pure-signal endpoints of about 4.0 to 4.4 pp for dropping the structural credit signal at $\alpha = 1$ against 2.3 to 2.8 pp for dropping the divergence signal at $\alpha = 0$. The structural credit signal is therefore the larger single-signal contributor.

Table 3: ALFWorld and WebShop success rate in percent for Qwen2.5-7B-Instruct across the signal-mixing coefficient α.

α	0	0.2	0.5	0.7	1.0
ALFWorld	89.1	90.6	**91.9**	87.5	87.5
WebShop	81.3	79.7	**83.6**	76.6	79.6

4.4 Analysis

Dependency graph structure over training. Figure 3 (left) tracks how GraphOPD reshapes the dependency graph over training, testing whether centrality-weighted supervision pushes the policy toward shorter, more decisive trajectories rather than merely re-weighting its existing behavior. It does, since the average number of graph nodes, equivalently trajectory length, and the average number of edges both fall over the first 150 steps, from about 47 and 39 at initialization to a plateau around 25 to 30, while graph density rises correspondingly from about 0.03 to between 0.10 and 0.14, consistent with the policy reaching the goal through fewer, more decisive actions rather than the same actions simply being scored differently.

Credit-content audit. Table 4 validates the structural credit signal. The ablations show that the structural credit signal improves outcomes, and the audit tests directly whether it tracks the steps that decide the outcome, via executed replay (Zhang, 2026). For 30 trajectories of the trained 7B policy we sample 10 decision steps per trajectory, re-sample 4 alternative actions at each, roll forward to the horizon, and take the induced change in success as ground-truth impact. For each signal we report the Spearman correlation ρ with this impact and Hit@10%, the fraction of its top-10% predicted steps that are truly

Table 4: **Credit-content audit on ALFWorld at 7B.** Metrics defined in the text.

Signal	ρ	Hit@10%
ℓ_{graph}	**0.58**	**0.48**
ℓ_{model}	0.34	0.41
Random	-0.06	0.13

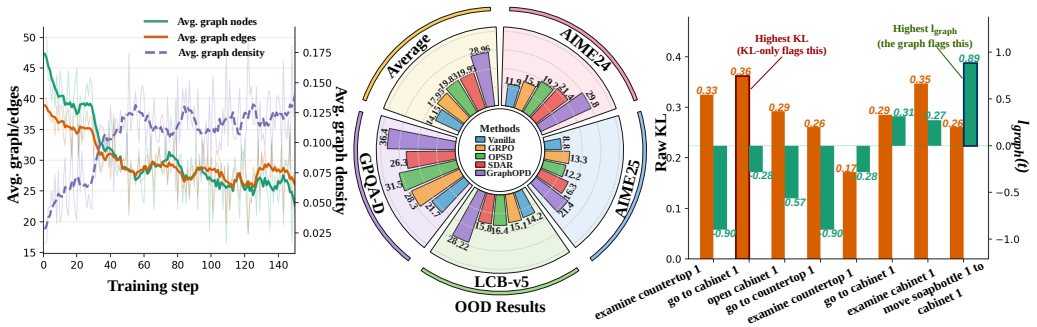


Figure 3: **Analysis.** (**Left**) Average graph nodes, equivalently trajectory length, average edges, and average density of the dependency graphs built on ALFWorld at 7B over the 150 training steps; nodes and edges fall from about 47 and 39 to a plateau of 25 to 30 while density rises from about 0.03 to between 0.10 and 0.14, so trajectories grow shorter and denser. (**Middle**) Accuracy radar on the four out-of-domain tool-integrated-reasoning benchmarks, AIME24 and AIME25 under avg@32 and LiveCodeBench-v5 and GPQA-Diamond under avg@8, plus their average, for GraphOPD against Vanilla, GRPO, OPSD, and SDAR; the GraphOPD polygon dominates every axis, averaging 29.0% against 20.0% for SDAR. Figure 3 (right) makes the mechanism concrete on a fragment of one real successful ALFWorld trajectory, where raw KL peaks at a cabinet revisit that never decides the outcome while ℓ_{graph} peaks at the decisive object-placement step within this fragment.

top-10% impactful, for which chance is 0.10. ℓ_{graph} scores $\rho = 0.58$ with Hit@10% $= 0.48$, clearly above both the random baseline at $\rho = -0.06$ and Hit@10% $= 0.13$ and ℓ_{model} at $\rho = 0.34$ and Hit@10% $= 0.41$. The structural credit signal thus identifies outcome-relevant steps more reliably than the divergence gap alone, and the Shuffled Centrality control isolates this same gain from mere non-uniform weighting.

Generalization and case study. Figure 3 (middle) tests whether the dependency-graph mechanism transfers beyond the three agentic benchmarks to a different domain and tool, on AIME24, AIME25, LiveCodeBench-v5, and GPQA-Diamond run as tool-integrated reasoning in mathematics, science, and code against Vanilla, GRPO, OPSD, and SDAR under the same backbone and Qwen2.5-7B-Instruct, and scored by avg@32 on the two AIME sets and avg@8 on the other two. GraphOPD attains the best accuracy on all four, averaging 29.0% against 20.0% for SDAR and 19.8% for OPSD, a +9.0 pp average gain over SDAR that reaches +12.4 pp on LiveCodeBench-v5; its lead is the one pattern holding on every benchmark, showing that the structural signal transfers where weighting by distributional deviation alone does not. Prompts, the Footprint extractor, and the full result table are in Appendix D. Figure 3 (right) makes the mechanism concrete on a fragment of one real successful ALFWorld trajectory, where raw KL peaks at a cabinet revisit that never decides the outcome while ℓ_{graph} peaks at the decisive object-placement step within this fragment.

5 Conclusion

We presented **GraphOPD**, which reads a per-step structural credit score from the trajectory and fuses it with the divergence signal to decide where a teacher's guidance is usable, distilling only each rollout's above-average-aptitude steps. Across three model scales, eleven baselines, and three agentic benchmarks spanning embodied control, web navigation, and retrieval-augmented reasoning, GraphOPD attains the best result on ALFWorld and WebShop at every scale, with a margin over the strongest baseline that reaches **+5.8 pp** on ALFWorld and **+5.5 pp** on WebShop, and the same structural signal proves effective on SearchQA as well, extending the approach to a substantially different action space. Ablations confirm that both signals are independently necessary, with the structural credit signal the larger contributor. Using an LLM to extract the Footprint directly from raw trajectories, rather than through hand-specified parsing rules, is a natural next step toward applying GraphOPD to a still broader range of agentic domains.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of portions of the manuscript, including revising selected passages for clarity and readability. Additionally, we used generative AI tools for retrieval and discovery, e.g., finding and screening related work and prior publications relevant to this study. We did not use generative AI tools for any other applicable tasks requiring disclosure beyond those stated here. We have reviewed all AI-assisted work. All AI-assisted textual revisions were checked by the authors to ensure that they preserved the intended meaning and accurately reflected the work, and every candidate reference surfaced through AI-assisted retrieval was independently located, read, and verified against its original source before being cited. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All benchmarks used in this work are publicly available. All the training code is released at the URL given in the abstract and included in the Supplementary Material; the method, hyperparameters, baseline reproduction details, prompts, and audit protocol are fully specified in the paper and its appendix.

REFERENCES

Chishui Chen, Yaoyou Fan, Te Sun, Yi Yang, Chenghao Sun, Delin Mao, Hongbo Qiao, Zuowei Zhang, Junxi Wang, Chenxing Sun, Yangen Hu, Lu Pan, Xuyang Liu, and Linfeng Zhang. Look ahead before you distill: Future trajectory validation of teacher guidance for agentic on-policy distillation. *CoRR*, abs/2608.01953, 2026. doi: 10.48550/ARXIV.2608.01953. URL https://doi.org/10.48550/arXiv.2608.01953.

Liyi Chen, Zhi Li, Yijun Wang, Tong Xu, Zhefeng Wang, and Enhong Chen. MMEA: entity alignment for multi-modal knowledge graph. In Gang Li, Heng Tao Shen, Ye Yuan, Xiaoyang Wang, Huawen Liu, and Xiang Zhao (eds.), *Knowledge Science, Engineering and Management - 13th International Conference, KSEM 2020, Hangzhou, China, August 28-30, 2020, Proceedings, Part I*, volume 12274 of *Lecture Notes in Computer Science*, pp. 134–147. Springer, 2020. doi: 10.1007/978-3-030-55130-8_12. URL https://doi.org/10.1007/978-3-030-55130-8_12.

Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. Multi-modal siamese network for entity alignment. In Aidong Zhang and Huzefa Rangwala (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 118–126. ACM, 2022. doi: 10.1145/3534678.3539244. URL https://doi.org/10.1145/3534678.3539244.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/4254e856d01a5e7b7ea050477c3ef9b9-Abstract-Conference.html.

Cristiano da Costa Cunha, Ajmal Mian, Tim French, and Wei Liu. Causal reinforcement learning for complex card games: A magic the gathering benchmark. *CoRR*, abs/2605.06066, 2026. doi: 10.48550/ARXIV.2605.06066. URL https://doi.org/10.48550/arXiv.2605.06066.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=uZbZb058bA.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.580. URL https://doi.org/10.18653/v1/2020.coling-main.580.

Ijun Jang, Je Won Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 42217–42227. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.2094. URL https://doi.org/10.18653/v1/2026.findings-acl.2094.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguistics*, 6:317–328, 2018. doi: 10.1162/TACL_A_00023. URL https://doi.org/10.1162/tacl_a_00023.

Sijia Li, Yuchen Huang, Zifan Liu, Yanping Li, Jingjing Fu, Li Zhao, Jiang Bian, Ling Zhang, Jun Zhang, and Rui Wang. GEAR: granularity-adaptive advantage reweighting for LLM agents via self-distillation. *CoRR*, abs/2605.11853, 2026a. doi: 10.48550/ARXIV.2605.11853. URL https://doi.org/10.48550/arXiv.2605.11853.

Wenwu Li, Yuran Song, Mingze Zhao, Bo Jin, and Wenhao Li. Unifying temporal and structural credit assignment in llm-based multi-agent prompt optimization. *CoRR*, abs/2605.30227, 2026b. doi: 10.48550/ARXIV.2605.30227. URL https://doi.org/10.48550/arXiv.2605.30227.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026c. URL https://openreview.net/forum?id=aaDDXXDGXB.

Kun Liang, Chenming Tang, Clive Bai, Weijie Liu, Saiyong Yang, and Yunfang Wu. ADWIN: adaptive windows for horizon-aware on-policy distillation. *CoRR*, abs/2605.28396, 2026. doi: 10.48550/ARXIV.2605.28396. URL https://doi.org/10.48550/arXiv.2605.28396.

Junzhuo Liu, Weiwei Li, Jun Ling, and Peng Wang. When privileged guidance misaligns: State-matched routing and contextualized self-distillation for multi-turn agents. *CoRR*, abs/2608.05219, 2026. doi: 10.48550/ARXIV.2608.05219. URL https://doi.org/10.48550/arXiv.2608.05219.

Zhengxi Lu, Zhiyuan Yao, Zhu Han, Zihan Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning. *CoRR*, abs/2605.15155, 2026. doi: 10.48550/ARXIV.2605.15155. URL https://doi.org/10.48550/arXiv.2605.15155.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

Carl Dean Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000. ISBN 978-0-898714-54-8. doi: 10.1137/1.9780898719512. URL https://my.siam.org/Store/Product/viewproduct/?ProductId=971.

Nirit Nussbaum-Hoffer, Nitay Calderon, Liat Ein-Dor, and Roi Reichart. LLM explainability with counterfactual chains and causal graphs. *CoRR*, abs/2606.05972, 2026. doi: 10.48550/ARXIV.2606.05972. URL https://doi.org/10.48550/arXiv.2606.05972.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 5687–5711. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.378. URL https://doi.org/10.18653/v1/2023.findings-emnlp.378.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://dl.acm.org/doi/10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *Proceedings of the International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *CoRR*, abs/2604.00626, 2026. doi: 10.48550/ARXIV.2604.00626. URL https://doi.org/10.48550/arXiv.2604.00626.

Changhui Sun, Lanbo Liu, Hang Lei, Tong Ling, Jiahang Xie, Zhiyong Zheng, Yujia Wang, Hao Liu, Feng Xiao, Lu Liu, Yanlong Du, Zifeng Cheng, Ziwei Jiang, and Qing Gu. Step-level on-policy distillation: Interpolating between on-policy distillation and supervised fine-tuning. *CoRR*, abs/2608.16333, 2026. doi: 10.48550/ARXIV.2608.16333. URL https://doi.org/10.48550/arXiv.2608.16333.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/tacl_a_00475.

Hanyang Wang, Weijieying Ren, Yuxiang Zhang, Ding Cao, Zhizhao Zeng, Ke Zeng, and Tianxiang Zhao. Bipace: Bisimulation-guided policy optimization with action counterfactual estimation for LLM agents. *CoRR*, abs/2606.25556, 2026a. doi: 10.48550/ARXIV.2606.25556. URL https://doi.org/10.48550/arXiv.2606.25556.

Hao Wang, Lei Sha, and Jie M. Zhang. Stepcodereasoner: Aligning code reasoning with stepwise execution traces via reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=0vTONq9Z8j.

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. *CoRR*, abs/2604.10674, 2026c. doi: 10.48550/ARXIV.2604.10674. URL https://doi.org/10.48550/arXiv.2604.10674.

Jianze Wang, Ying Liu, Jinlong Chen, Xuchun Hu, Qilong Zhang, Yu Cao, Jun Wang, Hua Yang, Yong Xie, and Qianglong Chen. Mad-opd: Breaking the ceiling in on-policy distillation via multi-agent debate. *arXiv preprint arXiv:2605.01347*, 2026d.

Jiaqi WANG, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=KH9eJRz8WR.

Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 721–729. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/wang21b.html.

Qiuchen Wang, Shihang Wang, Yu Zeng, Qiang Zhang, Fanrui Zhang, Zhuoning Guo, Bosi Zhang, Wenxuan Huang, Lin Chen, Zehui Chen, Pengjun Xie, and Ruixue Ding. Navigating massive visual context in retrieval-augmented generation via multimodal memory graph. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=fylE5O8g5F.

Xucong Wang, Ziyu Ma, Yong Wang, Yuxiang Ji, Shidong Yang, Guanhua Chen, Pengkun Wang, and Xiangxiang Chu. APPO: agentic procedural policy optimization. *CoRR*, abs/2606.12384, 2026b. doi: 10.48550/ARXIV.2606.12384. URL https://doi.org/10.48550/arXiv.2606.12384.

Zechuan Wang, Siyuan Lu, Hongxuan Zhang, Linjian Mo, Chenyi Zhuang, and Leilei Gan. Teach the magnitude, not the direction: Verifier-bounded credit assignment for multi-turn multi-step LLM agents. *CoRR*, abs/2608.13179, 2026c. doi: 10.48550/ARXIV.2608.13179. URL https://doi.org/10.48550/arXiv.2608.13179.

Shiwei Wu, Joya Chen, Tong Xu, Liyi Chen, Lingfei Wu, Yao Hu, and Enhong Chen. Linking the characters: Video-oriented social graph generation via hierarchical-cumulative GCN. In Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (eds.), *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pp. 4716–4724. ACM, 2021. doi: 10.1145/3474085.3475684. URL https://doi.org/10.1145/3474085.3475684.

Wei Wu, Chao Wang, Dazhong Shen, Chuan Qin, Liyi Chen, and Hui Xiong. AFDGCF: adaptive feature de-correlation graph collaborative filtering for recommendations. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 1242–1252. ACM, 2024. doi: 10.1145/3626772.3657724. URL https://doi.org/10.1145/3626772.3657724.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *CoRR*, abs/2604.03128, 2026a. doi: 10.48550/ARXIV.2604.03128. URL https://doi.org/10.48550/arXiv.2604.03128.

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. *CoRR*, abs/2606.26790, 2026b. doi: 10.48550/ARXIV.2606.26790. URL https://doi.org/10.48550/arXiv.2606.26790.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *CoRR*, abs/2602.12125, 2026c. doi: 10.48550/ARXIV.2602.12125. URL https://doi.org/10.48550/arXiv.2602.12125.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

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. 16810–16824, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Jiazhen Yuan, Wei Zhao, and Zhengbiao Bai. Graph-enhanced policy optimization in LLM agent training. *CoRR*, abs/2510.26270, 2025. doi: 10.48550/ARXIV.2510.26270. URL https://doi.org/10.48550/arXiv.2510.26270.

Zike Yuan, Yukun Cao, Han Zhang, Jianzhi Yan, Le Liu, Cai Ke, Yue Yu, Hui Wang, Ming Liu, and Bing Qin. Egl-sca: Structural credit assignment for co-evolving instructions and tools in graph reasoning agents. *arXiv preprint arXiv:2605.10366*, 2026.

Haiyue Zhang. Credit without ground truth: Auditing step-level credit assignment in LLM agents against executed replay. *CoRR*, abs/2608.19760, 2026. doi: 10.48550/ARXIV.2608.19760. URL https://doi.org/10.48550/arXiv.2608.19760.

Huan Zhang, Mingju Chen, Dongxu Zhou, Can Lv, Heng Chang, Sen Cui, Faguo Wu, and Shiji Zhou. TRCA: transition-wise rubric credit assignment for long-horizon LLM agents. *CoRR*, abs/2608.16156, 2026a. doi: 10.48550/ARXIV.2608.16156. URL https://doi.org/10.48550/arXiv.2608.16156.

Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. Temporal graph contrastive learning for sequential recommendation. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 9359–9367. AAAI Press, 2024. doi: 10.1609/AAAI.V38I8.28789. URL https://doi.org/10.1609/aaai.v38i8.28789.

Yanfei Zhang, Xu Lin, and Chenglin Wu. Stepopsd: Step-aware online preference self-distillation for agent reinforcement learning, 2026b. URL https://arxiv.org/abs/2605.27140.

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. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. SOD: step-wise on-policy distillation for small language model agents. *CoRR*, abs/2605.07725, 2026. doi: 10.48550/ARXIV.2605.07725. URL https://doi.org/10.48550/arXiv.2605.07725.

A Footprint Extractor and Algorithm

Each environment has a dedicated, rule-based Footprint extractor $\mathcal{F}_{\text{env}}$ that maps an action and observation to a small set of `<type>:<entity>` keys. The extractor for the four out-of-domain tool-integrated-reasoning benchmarks is specified in Appendix D.

Footprint as a state-difference projection. $\mathcal{F}_{\text{env}}$ is the practical, text-only form of a more general construction, given here in full. Treat the environment state s_t from Section 3.1 as a set of ground predicates over named entities, for instance `location(apple_1)=countertop_1`, so that the raw effect of step t is the symmetric difference $\Delta_t = s_t \triangle s_{t-1}$, namely the predicates that came into or out of existence, or changed value, when action a_t was executed against observation o_t. Most of Δ_t is irrelevant to any later step, because the specific new value a predicate takes rarely determines whether a downstream action depends on it, whereas the fact that some named entity changed does. We therefore canonicalize Δ_t through a projection π that retains only the `<type>:<entity>` pair each changed predicate is indexed by and discards its value,

$$\mathcal{F}(t) = \pi(\Delta_t) = \mathcal{F}_{\text{env}}(a_t,\, o_t). \tag{12}$$

$\mathcal{F}_{\text{env}}$ is implemented directly from (a_t, o_t) rather than from a structured s_t, since in a text-interfaced environment the action and observation text are exactly what fixes Δ_t in the first place.

Lemma A.1 (Footprint sufficiency)**.** *$\mathcal{F}(t)$ is a sufficient statistic of Δ_t for the dependency-edge relation E of Eq. equation 2. For any two trajectories whose step-t effects Δ_t, Δ'_t project to the same Footprint, $\pi(\Delta_t) = \pi(\Delta'_t)$, the edge test returns the same decision for every later step j.*

Proof. Eq. equation 2 decides $(v_t \to v_j) \in E$ by testing whether some `<type>:<entity>` $\in \mathcal{F}(t)$ has its `<entity>` occur as a substring of a_j. This test reads Δ_t only through $\pi(\Delta_t) = \mathcal{F}(t)$, and it never inspects a predicate's value, i.e. never reads anything in $\Delta_t \setminus \pi(\Delta_t)$. Two effects with identical $\pi(\Delta_t)$ therefore agree on every entity-membership query the test can make, and so yield an identical decision for every j, regardless of how Δ_t and Δ'_t otherwise differ. □

The projection π is lossy in general, discarding exactly the information Lemma A.1 shows the graph never needed. $\mathcal{F}(t)$ is therefore not merely a convenient summary of Δ_t but the coarsest one that still determines E.

Table 5 lists the action types and Footprint key formats used for each of the three benchmarks, and none of these extractors involve an LLM call, semantic embedding, or privileged annotation.

As a concrete example, suppose step 2 of an ALFWorld trajectory is `take apple 1 from countertop 1`, yielding `holding:apple 1`, and step 5 is `put apple 1 in fridge`

Table 5: Footprint extractor specifications by environment.

Environment	Key action types	Footprint key format
ALFWorld	go to, take, put, heat/cool/clean, open/close	`location:<loc>`, `visible_at:<loc>`, `holding:<obj>`, `placed:<obj>@<loc>`, `object_state:<obj>:<prop>`, `container_state:<container>`
WebShop	search[q], click[product_id], click[attr], click[buy]	`query_token:<word>`, one per query word, `selected_product:<id>`, `selected_attr:<attr>`, `purchase_action`
SearchQA	`<search>`, `<think>`, `<answer>`	`query_token:<word>`, `retrieved_passages`, `reasoning_state`, `answer_candidate`, `passage_entity:<entity>`

`1`, whose action text contains `apple 1`. A dependency edge $v_2 \to v_5$ is then added with weight $1/df(\texttt{apple 1})$ following Eq. equation 3. Analogously, a WebShop search produces one `query_token` key per query word, and a later action whose text contains one of those words draws the corresponding edge.

Complete algorithm. Algorithm 1 summarizes the full GraphOPD pipeline for one trajectory, from Footprint extraction to the combined training objective. It is a drop-in addition to the GRPO loop, since the graph, the two signals, the aptitude, and the mask are all computed from the rollout and its recorded states, with no extra model calls.

Algorithm 1 GraphOPD, per-trajectory supervision weighting

Require: trajectory $\tau = (s_0, a_1, \ldots, a_T, s_T)$, student π_θ, teacher $\pi_{\theta_{\text{old}}}$, coefficients α, λ, damping d

1: **Footprints.** For $t = 1, \ldots, T$, extract $\mathcal{F}(t) = \mathcal{F}_{\text{env}}(a_t, o_t)$

2: **Dependency graph.** Build $G = (V, E, w)$, and for each $i < j$ add $v_i \to v_j$ if an entity in $\mathcal{F}(i)$ is referenced by a_j, weighted $1/df(e)$ per Eq. equation 3, then add backtrack edges of weight 0.5 and sequential edges of weight 0.05

3: **Centrality.** Row-normalize w into M, giving leaf nodes a uniform row $1/T$, solve $\boldsymbol{\pi}^* = (1-d)\boldsymbol{\pi}^* M + d\mathbf{u}^\top$ by power iteration, and set $\ell_{\text{graph}}(t) = \log\big(T\,\pi^*_t\big)$

4: **Deviation.** $\delta(t) = \frac{1}{|S_t|}\sum_{k \in S_t}\big|\log \pi_{\theta_{\text{old}}}(k|s_t) - \log \pi_\theta(k|s_t)\big|$, then $r_t = \delta(t)/\delta_{\max}$ and $\ell_{\text{model}}(t) = \log\big[(r_t + \varepsilon)/(1 - r_t + \varepsilon)\big]$

5: **Aptitude.** $A(t) = \alpha\,\ell_{\text{model}}(t) + (1-\alpha)\,\ell_{\text{graph}}(t)$

6: **Mask.** $\mu_\tau, s_\tau \leftarrow$ mean, std of $\{A(t)\}_{t=1}^T$, then $m(t) = \sigma\big((A(t) - \mu_\tau)/s_\tau\big)$

7: **Objective.** $\mathcal{L} = \mathcal{L}_{\text{GRPO}} + \lambda \frac{\sum_t m(t)\,\text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t)\,\|\,\pi_\theta(\cdot|s_t))}{\sum_t m(t)}$

8: Update θ by gradient descent on $\mathcal{L}$

B Theoretical Properties and Implementation Notes

Two claims made informally in Section 3.4 are provable consequences of Eqs. equation 4 to equation 9 rather than free-standing design choices. This section states and proves them, gives a full derivation of the convergence rate they rely on, and draws three corollaries connecting that derivation to specific empirical results, namely the observed power-iteration count, the Soft-Pruning ablation, and the shrinking trajectory lengths in the dependency-graph-structure analysis.

The stationary distribution exists, is unique, and is efficiently computable. Eq. equation 4's update matrix $P = (1-d)M + d\,\mathbf{1}\mathbf{u}^\top$ combines the row-stochastic edge matrix M (row-normalized, with leaf nodes given the uniform row $1/T$) with the uniform teleportation matrix $\mathbf{1}\mathbf{u}^\top$ under weight $d \in (0, 1)$. Every entry of P is therefore at least $d/T > 0$, so P is a primitive, meaning irreducible

and aperiodic, row-stochastic matrix for *any* graph topology G, including disconnected components or absorbing states that would make M alone reducible. By the Perron–Frobenius theorem for primitive non-negative matrices (Meyer, 2000), P's spectral radius is a simple eigenvalue equal to 1, with a strictly positive left eigenvector unique up to scale, and normalizing $\sum_t \pi^*_t = 1$ fixes that scale, so $\boldsymbol{\pi}^*$ is well-defined regardless of G's structure and independent of the power-iteration starting point. The same damped-teleportation construction is the basis of the classical convergence guarantee for stationary distributions on directed graphs (Page et al., 1999), under which P's second-largest eigenvalue modulus is bounded by $1 - d$, so power iteration converges geometrically at rate $O((1-d)^k)$, matching the empirical convergence noted below (under 50 iterations at $d = 0.15$, $T \leq 50$, to tolerance 10^{-6}).

Proposition 1 (Bounds on the two signals)**.** *For any step t in a trajectory of length T with damping $d \in (0, 1)$,*

$$\log d \;\leq\; \ell_{\textit{graph}}(t) \;\leq\; \log T,$$

and for any smoothing floor $\varepsilon > 0$,

$$-\log \frac{1+\varepsilon}{\varepsilon} \;\leq\; \ell_{\textit{model}}(t) \;\leq\; \log \frac{1+\varepsilon}{\varepsilon}.$$

Proof. For ℓ_{graph}, since M is entrywise non-negative and $\boldsymbol{\pi}^*$ is non-negative, Eq. equation 4 gives $\boldsymbol{\pi}^* = (1-d)\boldsymbol{\pi}^* M + d\,\mathbf{u}^\top \geq d\,\mathbf{u}^\top$ entrywise, so $\pi^*_t \geq d/T$ for every t and $\ell_{\text{graph}}(t) = \log(T\pi^*_t) \geq \log d$. The upper bound follows because $\pi^*_t \leq 1$ trivially, as $\boldsymbol{\pi}^*$ is a probability distribution over T steps, giving $\ell_{\text{graph}}(t) \leq \log T$. For ℓ_{model}, $r_t \in [0, 1]$ by construction, so $\frac{r_t+\varepsilon}{1-r_t+\varepsilon} \in \left[\frac{\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{\varepsilon}\right]$, and taking logs gives the stated bound. □

With the values used throughout, $d = 0.15$ and $\varepsilon = 0.1$, this gives $\ell_{\text{graph}}(t) \in [-1.90, \log T]$ and $\ell_{\text{model}}(t) \in [-2.40, 2.40]$, and for trajectories capped at $T = 50$ steps, $\log T \leq 3.91$, so the two signals occupy overlapping ranges of comparable magnitude, the exact sense in which the two signals are commensurate.

Proposition 2 (Affine invariance of the trajectory-relative mask)**.** *Let $A'(t) = a \cdot A(t) + b$ for constants $a > 0, b \in \mathbb{R}$ applied uniformly across a trajectory τ. Then the mask computed from A' equals the mask computed from A, $m'(t) = m(t)$ for every $t \in \tau$.*

Proof. Let μ_τ, s_τ be the mean and standard deviation of $\{A(t)\}_{t=1}^T$. The corresponding statistics of $\{A'(t)\}$ are $\mu'_\tau = a\mu_\tau + b$ and $s'_\tau = a\,s_\tau$, since variance scales by a^2 and $a > 0$, so the standard deviation scales by a. Then

$$\frac{A'(t) - \mu'_\tau}{s'_\tau} = \frac{(aA(t) + b) - (a\mu_\tau + b)}{a\,s_\tau} = \frac{A(t) - \mu_\tau}{s_\tau},$$

so $m'(t) = \sigma\big(\frac{A'(t)-\mu'_\tau}{s'_\tau}\big) = \sigma\big(\frac{A(t)-\mu_\tau}{s_\tau}\big) = m(t)$. □

This is the precise sense in which the mask stays meaningful as $A(t)$'s absolute scale drifts over training. Any rescaling of the aptitude that is common across a trajectory's own steps leaves every mask weight in that trajectory unchanged, and such a common rescaling is exactly what happens as δ_{max} and the graph's stationary distribution shift with the policy. A threshold fixed across trajectories has no such guarantee, since Proposition 2 only holds for a single shared (a, b) within one trajectory, and different trajectories drift by different amounts, which is why a global threshold would need continual re-tuning while the trajectory-relative one does not.

A first-principles proof of the convergence rate. The Perron–Frobenius argument above is an existence and uniqueness statement, and it does not by itself derive the geometric rate $O((1-d)^k)$ used throughout the paper. The same rate follows from an elementary contraction argument that avoids any assumption on M's spectrum. For a row-stochastic matrix M and any real vector x,

$$\|xM\|_1 = \sum_j \Big|\sum_i x_i M_{ij}\Big| \leq \sum_j \sum_i |x_i| M_{ij} = \sum_i |x_i| \sum_j M_{ij} = \|x\|_1, \tag{13}$$

using only the triangle inequality and $\sum_j M_{ij} = 1$. Let $\boldsymbol{\pi}_k$ be the k-th power-iteration estimate and $\boldsymbol{\pi}^*$ the fixed point of Eq. equation 4. Subtracting $\boldsymbol{\pi}^* = (1-d)\boldsymbol{\pi}^* M + d\mathbf{u}^\top$ from $\boldsymbol{\pi}_{k+1} = (1-d)\boldsymbol{\pi}_k M + d\mathbf{u}^\top$ cancels the teleportation term exactly, leaving $\boldsymbol{\pi}_{k+1} - \boldsymbol{\pi}^* = (1-d)(\boldsymbol{\pi}_k - \boldsymbol{\pi}^*)M$, so applying Eq. equation 13 with $x = \boldsymbol{\pi}_k - \boldsymbol{\pi}^*$ and iterating gives

$$\|\boldsymbol{\pi}_k - \boldsymbol{\pi}^*\|_1 \;\le\; (1-d)^k \,\|\boldsymbol{\pi}_0 - \boldsymbol{\pi}^*\|_1 \;\le\; 2(1-d)^k, \tag{14}$$

where the last step uses that both are probability vectors, so their L_1 distance is at most 2 regardless of initialization or graph topology. Solving Eq. equation 14 for the smallest k that reaches a target tolerance ε_{tol} gives $k \ge \log(2/\varepsilon_{\text{tol}})/(-\log(1-d))$, and at $d = 0.15$ with the training tolerance $\varepsilon_{\text{tol}} = 10^{-6}$ used throughout, this bound is $k \approx 89$. The power iteration in practice converges in under 50 steps, roughly half of this worst-case figure, since Eq. equation 14 bounds the distance from an *arbitrary* initialization by the loosest possible constant 2, the diameter of the probability simplex in L_1, while power iteration is initialized at the uniform distribution $\mathbf{u}$ itself, which is already closer to $\boldsymbol{\pi}^*$ than a worst-case starting point.

Corollary on the edge-weighting scheme. Eq. equation 14's derivation uses only that M is row-stochastic, and it does not use how the pre-normalization edge weights $w(v_i \to v_j)$ were assigned. The soft-pruning weights $w = 1/df(e)$ of Eq. equation 3 and the Soft-Pruning ablation's constant weights (Table 2) both row-normalize to a valid M, so Proposition 1's existence, uniqueness, and convergence guarantees hold identically for both. The gap between them in Table 2, 0.8 to 5.5 pp in Section 4.3, is therefore attributable entirely to which entities the two weighting schemes make central, not to any difference in whether $\boldsymbol{\pi}^*$ is well-defined or how fast it is reached.

Corollary on signal range and trajectory length. Proposition 1 bounds $\ell_{\text{graph}}(t)$ within $[\log d,\ \log T]$, a range whose width is set entirely by the current trajectory's length T, equivalently the graph's node count, once d is fixed. Figure 3 (left) reports T contracting from about 47 nodes at initialization to a plateau of 25 to 30 over the first 150 training steps. By Proposition 1, this alone narrows the theoretical upper end of ℓ_{graph}'s range from $\log 47 \approx 3.85$ to between $\log 25 \approx 3.22$ and $\log 30 \approx 3.40$, while the lower end, $\log d \approx -1.90$, is unaffected by T. The structural credit signal therefore continues to operate over a range of comparable width as trajectories shorten, rather than collapsing toward the neutral point 0 as T falls. This is consistent with the ablations and the audit finding the structural credit signal informative throughout training rather than only early on.

Intermediate steps carry structural credit, not only terminal ones. Eq. equation 4 is a fixed point over all steps, not an accumulation at the trajectory's end: $\pi_t^* = (1-d)\sum_i \pi_i^* M_{it} + d/T$, so a step's mass is exactly the rarity-weighted inflow of the dependency chains its action consumes, plus the teleport floor. Accordingly $\ell_{\text{graph}}(t) > 0$ precisely when this inflow exceeds the uniform level $1/T$, which happens wherever several upstream state changes are exercised together, at intermediate milestones such as retrieving an object, opening or heating it, or carrying it to a target receptacle, and not only at the final action. Two properties keep this intermediate credit actionable: the mask of Eq. equation 9 standardizes aptitude within each trajectory, so a mid-trajectory milestone that outranks the resolving step in its own rollout receives the larger supervision weight, and Proposition 1 floors every step's mass at d/T, so no step is starved before masking. Empirically, the credit-content audit, which perturbs decision points across sampled trajectories, ranks them against executed-replay impact at $\rho = 0.58$ with Hit@10% $= 0.48$ against a chance level of 0.10.

The role of the damping factor. The damping factor d trades off two properties that Eq. equation 14 and Proposition 1 make explicit and opposing. A larger d tightens the convergence bound, $k = O(1/d)$ for small d, but also raises the floor $\log d$ toward 0, compressing ℓ_{graph}'s range and pulling $\boldsymbol{\pi}^*$ closer to the uninformative uniform distribution $\mathbf{u}$ in the limit $d \to 1$. A smaller d widens the usable range through a lower, more negative floor that lets peripheral steps register as clearly below neutral rather than merely below average, but it slows convergence and, in the degenerate limit $d \to 0$, forfeits the guarantee above that M alone is primitive on every graph topology. The selected $d = 0.15$ sits well inside this trade-off, since it is small enough that $\log d \approx -1.90$ leaves ℓ_{graph} a range comparable to ℓ_{model}'s ± 2.40 (Proposition 1), while large enough that the 89-iteration worst-case bound above remains a small fraction of a single training step's compute budget.

Graph construction cost. Graph construction is $O(T^2|\mathcal{F}|)$ per trajectory, negligible at typical agent trajectory lengths, $T \le 50$. Table 6 measures this directly, comparing the wall-clock time

spent on Footprint extraction, edge construction, and the stationary-distribution solve against the total per-training-step time, where graph construction accounts for well under half a percent of the step budget on every benchmark.

Table 6: Wall-clock cost of graph construction relative to the total per-training-step time.

Benchmark	Graph construction (s/step)	Total step time (s)	Share
ALFWorld	0.6659	342.0936	0.19%
WebShop	0.4535	144.2885	0.31%
SearchQA	0.4144	322.2142	0.13%

Centrality computation. The damping term $d > 0$ guarantees a unique stationary distribution regardless of graph topology, by the Perron–Frobenius argument given above, and it is obtained by power iteration to tolerance $\|\boldsymbol{\pi}^*_{k+1} - \boldsymbol{\pi}^*_k\|_1 < 10^{-6}$, which converges in under 50 iterations for $T = 50$, consistent with the $O((1-d)^k)$ geometric rate proved there.

Numerical details. The floor $\varepsilon = 0.1$ in Eq. equation 7 bounds the divergence signal and keeps the log-odds finite at $r_t \in \{0, 1\}$, and Proposition 1 gives the exact range of both signals and why neither can unilaterally dominate the fusion. Because $\sigma(\cdot)$ lies strictly between 0 and 1 for any finite argument, the mask sum $\sum_t m(t)$ is positive on every trajectory with $s_\tau > 0$. On the rare trajectory where $A(t)$ is exactly constant across all steps, so that $s_\tau = 0$, no step's aptitude differs from the trajectory's own average, so we set $m(t) = \sigma(0) = \frac{1}{2}$ for all t, i.e. uniform weighting. Either way, the normalized loss in Eq. equation 10 is always a well-defined weighted average, and the normalization keeps the loss magnitude invariant to mask sparsity, so λ needs no re-tuning as sparsity shifts over training.

Soft mask. Unlike hard-threshold masking, the soft mask varies smoothly with the standardized aptitude, reallocating the supervision budget gradually rather than admitting or excluding a step outright, and it is computed from the rollout's detached statistics, acting as a fixed per-step weight within each update. Standardizing within the trajectory before the sigmoid, rather than applying one threshold across all trajectories, keeps the same relative cutoff, above this trajectory's own average aptitude, meaningful regardless of where that trajectory's absolute aptitude scale happens to sit at that point in training (Proposition 2).

Effective concentration of the soft mask. Read literally, the mask of Eq. equation 9 never excludes a step: σ maps every finite aptitude into $(0, 1)$, so every step retains a positive weight and the normalized loss of Eq. equation 10 is a re-weighting of the distillation term rather than a hard selection of the kind probed in Figure 1. The relevant question is therefore how strongly the re-weighting concentrates, and the standardized form of the mask makes this quantifiable: since $z(t) = (A(t) - \mu_\tau)/s_\tau$ is exactly zero-mean and unit-variance within each trajectory, the concentration of $m(t) = \sigma(z(t))$ depends only on the shape of the trajectory's aptitude distribution. Under a Gaussian reference shape the effective sample size $\mathrm{ESS} = (\sum_t m(t))^2 / \sum_t m(t)^2$ equals about $0.85\,T$ and the above-mean mass share $\sum_{z>0} m(t) / \sum_t m(t)$ about 0.68; on the two logged rollouts used in our case studies, a 50-step and a 6-step episode, the same quantities compute to 0.85 with share 0.67 and 0.85 with share 0.66, the mean weight of above-mean steps exceeds that of below-mean steps by about a factor of two, and extreme steps differ by up to a factor of seven ($m \in [0.13, 0.93]$ on the 50-step rollout), whose twelve highest-aptitude steps absorb 37% of the distillation mass under the soft mask against 100% under a budget-matched hard top-12 selection. Three considerations make this moderate concentration the intended operating point rather than a shortfall. First, the hard top-k rule of Figure 1 is a diagnostic of the divergence signal, not a candidate training mask: it was applied to raw KL precisely to test whether divergence magnitude locates outcome-relevant steps, so it shares neither the aptitude's placement nor the smoothness properties below. Second, the KL term against $\pi_{\theta_{\text{old}}}$ also acts as an on-support trust region; a hard mask would remove that anchor from the majority of steps, whereas the soft mask keeps a positive floor $m(t) \geq \sigma(\min_{t'} z(t')) > 0$ on every step. Third, concentration without correct placement is worthless: the Shuffled Centrality control of Table 2 redistributes exactly the same mask values, hence exactly the same dose and the same ESS, onto the wrong steps and loses up to 6.2 pp, so the

active ingredient is where the mass goes rather than how hard the selection is; a budget-matched hard-mask variant of GraphOPD is a natural ablation left to future work, and the evidence above indicates it would alter the dose distribution without altering the placement that the ablation identifies as decisive.

Teacher cost. Using $\pi_{\theta_{\text{old}}}$ as the teacher requires no additional model, checkpoint, or inference call beyond the GRPO training loop, and the drop-in wrapper leaves the GRPO optimizer and rollout infrastructure unmodified.

C Experimental Setup and Motivating Study Details

Benchmarks. This paragraph gives each of the three agentic benchmarks' task format, dependency structure, and evaluation protocol. ALFWorld (Shridhar et al., 2021) embeds a text interface around simulated household scenes, requiring an agent to retrieve, heat, clean, or relocate objects through a fixed vocabulary of navigation and manipulation actions (Table 5). Almost every action changes which objects and locations are reachable next, so a single early misstep can make the goal unreachable no matter how the remainder of the trajectory is executed, and this is where the dependency-graph hypothesis makes its strongest prediction. We score it by task success rate on the standard 3,553/140 train/test split, with episodes capped at 50 turns. WebShop (Yao et al., 2022) simulates an e-commerce search-and-purchase task over a large product catalog, with an action space of free-form search queries and page-element clicks rather than ALFWorld's fixed verb-object grammar. Its dependencies are informational rather than physical, since later clicks depend on which attributes and product identifiers an earlier search surfaced, so the Footprint extractor tracks query tokens and selected products instead of object locations. We report both completion score and binary success rate on the standard 1,012/500 split, capped at 15 turns. SearchQA aggregates seven open-domain question-answering datasets, categorized into single-hop factoid questions, NarrativeQA (Kociský et al., 2018), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023), and multi-hop questions, HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023), formatted as a `<search>`/`<think>`/`<answer>` loop capped at 10 turns. Here dependencies run only through retrieved evidence, with no physical state change at all, which tests whether the mechanism generalizes beyond the setting that motivated it, and we score it by accuracy over the seven datasets. The ALFWorld and SearchQA Avg columns of Table 1 weight each per-category number by that category's episode count in the evaluation split, so they do not reduce to the unweighted mean of the displayed numbers. The four out-of-domain benchmarks probed in Section 4.4 are specified together with their prompts, Footprint extractor, and full results in Appendix D.

Training configuration. All methods are optimized with AdamW at learning rate 10^{-6} under a cosine schedule with a 50-step warmup, batch size 64 trajectories per step formed as 8 tasks with 8 rollouts each, and gradient clipping at 1.0. GRPO uses group size $G = 8$ and clip ratio $\varepsilon = 0.2$. Each training run uses 8 H100 GPUs. Training runs for 150 steps on ALFWorld, WebShop, and SearchQA, with evaluation every 10 steps. GraphOPD's own hyperparameters are listed with the grid search in the final appendix section below, namely the mixing coefficient α, the loss coefficient λ, the centrality damping factor $d = 0.15$, its power-iteration tolerance 10^{-6}, and the smoothing floor $\varepsilon = 0.1$ of Eq. equation 7.

Run variance. GraphOPD, SDAR, OPID, and GEPO are each trained with three independent runs, and Table 1 reports their means on the three headline metrics, namely the ALFWorld average success rate, the SearchQA average accuracy, and the WebShop success rate. Table 7 lists the corresponding run-to-run standard deviations. GraphOPD has the smallest worst-case spread of the four, with a maximum standard deviation of 1.60 pp, while the largest is SDAR's 5.09 pp on 1.7B ALFWorld. All remaining rows of Table 1 are single-run point estimates.

Baseline families. The eleven baselines span four axes of comparison. Vanilla and GRPO establish lower bounds without any teacher signal, isolating how much of GraphOPD's gain is attributable to distillation at all rather than to the GRPO backbone alone. Skill-Prompt, Skill-GRPO, and Skill-SD inject external skill demonstrations or privileged information, and they therefore bound what such context adds on top of the same backbone. OPSD, GRPO+OPSD, RLSD (Yang et al.,

Table 7: Mean and run-to-run standard deviation (pp) over three independent runs for the four multi-run methods of the main results, on ALFWorld and WebShop success rate and SearchQA accuracy.

	Qwen3-1.7B			Qwen2.5-3B			Qwen2.5-7B		
Method	ALFWorld	SearchQA	WebShop	ALFWorld	SearchQA	WebShop	ALFWorld	SearchQA	WebShop
GraphOPD	$63.3_{\pm 1.18}$	$42.9_{\pm 0.05}$	$73.4_{\pm 1.60}$	$88.6_{\pm 0.57}$	$46.1_{\pm 0.02}$	$78.9_{\pm 0.43}$	$91.9_{\pm 0.14}$	$47.3_{\pm 0.40}$	$83.6_{\pm 0.43}$
SDAR	$51.3_{\pm 5.09}$	$40.7_{\pm 0.72}$	$59.9_{\pm 0.93}$	$81.8_{\pm 3.81}$	$43.4_{\pm 0.03}$	$69.3_{\pm 0.91}$	$84.4_{\pm 1.60}$	$47.5_{\pm 1.21}$	$82.0_{\pm 0.38}$
OPID	$58.2_{\pm 1.85}$	$39.6_{\pm 0.78}$	$66.4_{\pm 2.99}$	$82.8_{\pm 1.50}$	$44.6_{\pm 0.11}$	$74.5_{\pm 0.14}$	$89.1_{\pm 0.48}$	$48.2_{\pm 0.74}$	$80.0_{\pm 1.00}$
GEPO	$58.1_{\pm 2.58}$	$40.1_{\pm 0.38}$	$67.9_{\pm 1.20}$	$80.7_{\pm 3.84}$	$44.6_{\pm 1.91}$	$75.3_{\pm 1.35}$	$85.9_{\pm 5.02}$	$45.5_{\pm 2.85}$	$80.0_{\pm 1.85}$

2026a), SDAR (Lu et al., 2026), and OPID (Yang et al., 2026b) share GraphOPD's GRPO backbone and model family and allocate their distillation or advantage signal by teacher-student divergence magnitude alone, so they form our direct comparison set and the basis for the per-scale deltas in Section 4.2. GEPO (Yuan et al., 2025) instead reweights the GRPO advantage itself with a graph-based credit score rather than adding a distillation loss, and is the closest prior baseline to our own structural credit signal.

Prompts. Each benchmark's policy is queried with a fixed system prompt that fixes the admissible action vocabulary, and the Footprint extractor specified below parses exactly this vocabulary, so the prompt and the extractor are specified against the same action set by construction. Each prompt is instantiated once per episode with the task instruction and then re-issued every step with the running observation and action history appended. None of the three mentions the teacher, the dependency graph, or GraphOPD's training objective, so the graph and centrality computation (Section 3.2) have no visibility into or influence on what the policy is shown. The prompts for the four out-of-domain benchmarks are given in Appendix D.

```
ALFWorld. You are a household robot completing a task
in a simulated home. You will be given a task and, at
each step, an observation of the room. Choose exactly
one admissible action per step from: go to {receptacle},
open {receptacle}, close {receptacle}, take {object} from
{receptacle}, put {object} in/on {receptacle}, heat {object}
with {receptacle}, cool {object} with {receptacle}, clean
{object} with {receptacle}, examine {object}, use {object}.
Respond with exactly one line: "Action: {action}".

Task: {task_instruction}
Observation: {observation}
Admissible actions: {admissible_action_list}

WebShop. You are a shopping assistant browsing an online
store to complete a purchase instruction. At each turn you
see the current page and its clickable elements. Choose
exactly one action: search[{query}] to issue a search, or
click[{element}] to click a product ID, an attribute/option,
or the Buy Now button. Respond with exactly one line:
"Action: {action}".

Instruction: {purchase_instruction}
Page: {page_observation}
Clickable elements: {clickable_list}

SearchQA. Answer the question by interleaving reasoning and
retrieval. You may use <think>...</think> to reason and
<search>query</search> to retrieve passages, repeating
either as many times as needed, and must finish with
<answer>final answer</answer>.

Question: {question}
```

Motivating empirical study. The motivating study in Section 1 uses a 7B Qwen2.5-Instruct agent on ALFWorld under the same GRPO backbone and training configuration as the main experiments. For the selection comparison, per-trajectory distillation is restricted to the k highest-KL

steps, $k \in \{5, 8, 10, 12\}$, or to k randomly chosen steps, with all other conditions identical, and each configuration is trained for 150 steps. For the outcome-level check, 50 rollouts provide 466 successful and 466 failed steps.

D Out-of-Distribution Generalization

Benchmarks and protocol. Four out-of-domain benchmarks test whether the dependency-graph mechanism transfers beyond the three agentic settings to a different domain and a different tool. They are AIME24 and AIME25 in competition mathematics, GPQA-Diamond in graduate-level science, and LiveCodeBench-v5 in program synthesis, all run as multi-turn tool-integrated reasoning. AIME24 and AIME25 are scored by avg@32 and the other two by avg@8. Each benchmark uses a Python code interpreter and follows the protocol of SOD (Zhong et al., 2026), so the policy interleaves reasoning text with code-interpreter calls over several turns before committing to a final answer. Each turn therefore has an action, the model's own reasoning text or code, and an observation, the interpreter's response, which gives the four benchmarks the same action-observation structure as ALFWorld, WebShop, and SearchQA under a different tool and domain. This tests transfer across domains and tool vocabularies rather than across trajectory structure. All runs use Qwen2.5-7B-Instruct with the same GRPO backbone, training configuration, and step budget as the main experiments, and they compare against Vanilla, GRPO, OPSD, and SDAR.

Prompts. The two mathematics sets and GPQA-Diamond share the math/science template below, and LiveCodeBench-v5 uses the code template. Both are instantiated once per problem and re-issued every turn with the running reasoning text and interpreter history appended.

```
AIME24 / AIME25 / GPQA-Diamond. Analyze and solve the
following problem step by step. The code interpreter
can be called for precise or exploratory calculations and
to verify a result before committing to it; first form a
high-level solution strategy, then use the interpreter as
needed. Once the final answer is confirmed, stop calling
the interpreter and give it as \boxed{answer} on its own
line, an integer for AIME24/AIME25 and a single option
letter for GPQA-Diamond.
Problem: {problem}

LiveCodeBench-v5. Generate a correct Python program that
matches the following problem specification and passes
all tests. The code interpreter can be used to check the
program's correctness against the public examples before
submitting; first form a high-level solution strategy,
then use the interpreter to verify. The submitted program
must read from standard input and write to standard output,
enclosed in a single fenced code block.
Problem: {problem_statement}
Public examples: {public_examples}
```

Footprint extractor. The four benchmarks share one extractor, since a turn in each of them consists of reasoning text, an optional Python code block, and the interpreter's response, and the dependency structure runs through the identifiers and values that one turn introduces and a later turn reuses. Table 8 lists the keys that $\mathcal{F}_{\text{env}}(a_t, o_t)$ emits for such a turn, read from the turn's own code and reasoning text for the produced keys and from the interpreter response for the returned ones, in the same `<type>:<entity>` form as Table 5. Edge construction, rarity weighting, backtrack edges, and sequential edges then follow Eqs. equation 2 and equation 3 unchanged, with a_j the later turn's code and reasoning text and o_j the interpreter response, so a repeated identical interpreter output marks a revisit exactly as a repeated observation does in ALFWorld.

$df(e)$ counts the turns whose code, reasoning text, or interpreter response mentions e, so a helper variable introduced once and consumed by the next turn yields an edge weight near 1, while a loop counter or a common library name yields a small weight that the row normalization dilutes. The

Table 8: Footprint extractor specification for the four out-of-domain tool-integrated-reasoning benchmarks.

Source within the turn	Footprint key format
Code block and reasoning text	`symbol:<identifier>`, one per name bound by an assignment, a function definition, or a comprehension, `call:<function>`, one per interpreter function invoked, `import:<module>`, one per module imported, `answer_candidate:<value>`, for the value placed inside \boxed{} or submitted as the final program
Interpreter response	`result:<value>`, one per value printed or returned, `test_result:<example_id>`, one per public example checked on LiveCodeBench-v5

structural credit signal, the aptitude of Eq. equation 8, and the trajectory-relative mask of Eq. equation 9 carry over with the same α, λ, d, and ε as the agentic benchmarks, so the transfer experiment changes the domain and the tool vocabulary while holding the mechanism fixed.

Results. Table 9 reports the full numbers behind Figure 3 (middle), namely task accuracy on AIME24 and AIME25 under avg@32 and on LiveCodeBench-v5 and GPQA-Diamond under avg@8. GraphOPD attains the best accuracy on all four benchmarks, averaging 29.0% against 20.0% for SDAR and 19.8% for OPSD, and its lead holds on each benchmark separately, reaching +12.4 pp on LiveCodeBench-v5 over SDAR.

Table 9: Task accuracy (%) on four out-of-domain tool-integrated-reasoning benchmarks, with the scoring protocol per benchmark given in the text above.

Method	AIME24	AIME25	LCB-v5	GPQA-D	Avg
Vanilla	11.9	8.8	14.2	21.7	14.2
GRPO	15.1	13.3	15.1	28.3	18.0
OPSD	19.2	12.2	16.4	31.5	19.8
SDAR	21.4	16.3	15.8	26.3	20.0
GraphOPD	**29.8**	**21.4**	**28.2**	**36.4**	**29.0**

E Signal Validation and Case Study

Signal redundancy diagnostic. The fusion in Section 3.4 is a weighted sum of ℓ_{model} and ℓ_{graph}, so it implicitly assumes the two sources are not strongly *redundant*, since if they carried near-identical evidence, weighting both by α and $1-\alpha$ would double-count the same signal rather than combining two complementary ones. Rather than a separate correlation diagnostic, the ablation (Table 2, Section 4.3) already tests this functionally, because if ℓ_{model} and ℓ_{graph} carried near-identical evidence, dropping either one alone, Divergence-only or Structural-only, would cost little, as the remaining signal would recover most of the same information. Instead, both cost a substantial fraction of GraphOPD's full gain at every scale, Divergence-only 3.9 to 10.5 pp and Structural-only 2.3 to 6.5 pp, and neither recovers what the other drops, so the two signals are complementary rather than redundant.

Credit-content validation. The Analysis subsection of Section 4 reports the credit-content audit, and this appendix fixes its protocol.

Counterfactual audit. We construct causal ground truth on ALFWorld by executed replay, resampling the policy's own alternative action at each decision point of a sampled set of trajectories, rolling the environment forward from the resulting state, and recording whether the outcome changes, so a step's ground-truth contribution is the outcome change induced by its alternative, the same executed-replay principle used elsewhere to audit step-level credit signals (Zhang, 2026). We report, for each of ℓ_{graph}, ℓ_{model}, and a random score, the Spearman rank correlation ρ with this contribution and Hit@10%, the fraction of the signal's top-10% steps that are truly top-10% impactful,

for which chance is 0.10. The decisive comparison is ℓ_{graph} against the random score, since the claim that the structural credit signal identifies outcome-relevant steps requires both to sit clearly above chance, and the comparison against ℓ_{model} measures how much the structural credit signal adds over the divergence signal alone. ℓ_{graph} attains $\rho = 0.58$ and Hit@10% $= 0.48$, well above the random score at $\rho = -0.06$ and Hit@10% $= 0.13$ and ℓ_{model} at $\rho = 0.34$ and Hit@10% $= 0.41$.

Shuffled-centrality control. To separate credit content from supervision redistribution, the ablation (Table 2) includes a Shuffled Centrality row that replaces ℓ_{graph} with a within-trajectory permutation of itself, matched to the real one in mask sparsity and weight distribution, and runs the full training pipeline under otherwise identical conditions. Where the real graph retains its advantage over the shuffled one at matched dose, that gain is attributable to the content of the dependency structure rather than to the act of non-uniform weighting itself, and where it does not, part of the gain is attributable to redistribution alone. Table 2's discussion in the main text reports which of the two holds at each scale and benchmark.

Case study. Table 10 first gives a side-by-side rollout comparison on the same task and environment seed ("put some soapbottle on cabinet", `seed=1000`) under two checkpoints of the same training run, an early one at step 20 and a later one at step 150. Unlike a controlled trajectory pair that shares a fixed prefix, the two policies differ from the first action, so the table shows how far training reshapes the whole strategy rather than a single decision.

Table 10: A side-by-side comparison of two rollouts in ALFWorld. Both checkpoints attempt the same task and environment seed ("put some soapbottle on cabinet", `seed=1000`) from the same training run. Their strategies differ from the first action, so the early checkpoint never engages the target object at all.

Steps	**Late checkpoint, step 150 (succeeds)**	**Early checkpoint, step 20 (fails)**
0–2	**Action:** go to sinkbasin 1 **Action:** go to countertop 1 **Action:** take soapbottle 1 from countertop 1	**Action:** go to cabinet 1 **Action:** open cabinet 1 **Action:** examine cabinet 2
3–5 / 3–49	**Action:** go to cabinet 1 **Action:** open cabinet 1 **Action:** move soapbottle 1 to cabinet 1 **Result:** reward $= 10$, episode ends at step 5	**Actions (47 steps):** examine cabinet $\times 25$, look $\times 6$, open cabinet $\times 5$, go to cabinet $\times 3$, plus 8 other revisit actions, never mentioning the soapbottle **Raw KL:** ranges 0.16 to 0.65, mean 0.32 **Result:** reward $= 0$, times out at step 49
	The step-150 checkpoint carries the soapbottle from step 2 onward, so every later action follows directly from the object it already holds.	The step-20 checkpoint repeatedly opens and examines cabinets without ever picking up the soapbottle, so its failure is an entity it never engaged with rather than a late misstep.

Table 11 turns to the signal comparison from Figure 3 (right), an eight-step window of a single real successful ALFWorld trajectory ("put some soapbottle on cabinet"), with the raw per-step teacher-student KL and $\ell_{\text{graph}}(t)$ plotted as two aligned bar charts over the same eight steps. The table lists the same eight steps in full, giving the exact action text and both signals' values.

The decisive object-placement action at step 10, moving the soapbottle onto the cabinet, has one of the lowest raw KL values in the window, so a KL-only rule would spend its budget on step 4 instead, a revisit to the cabinet that does not bear on the outcome. ℓ_{graph} peaks exactly at step 10, since four earlier steps, step 4 (go to cabinet 1), step 5 (open cabinet 1), step 8 (go to cabinet 1 again), and step 9 (examine cabinet 1), all reference the same `cabinet 1` entity that step 10's action consumes. Both signals are computed by running the paper's own Footprint, edge-construction, and stationary-distribution pipeline on the real logged actions, using the dependency and sequential edges that the action text supports.

Table 12 adds a second case on the position of the credit peak, on a complete representative successful Heat-type episode (twelve steps, terminating at the resolving placement) whose actions and

Table 11: Qualitative case study on a window (steps 3 to 10) of a real successful ALFWorld trajectory ("put some soapbottle on cabinet"). Raw KL peaks at step 4, a cabinet-revisit step that does not decide the outcome, while $\ell_{\text{graph}}(t)$ peaks at step 10, the object-placement action decisive within this window. Bold marks each column's peak.

Step	Action	Raw KL	$\ell_{\text{graph}}(t)$
3	examine countertop 1	0.33	-0.90
4	go to cabinet 1	**0.36**	-0.28
5	open cabinet 1	0.29	-0.57
6	go to countertop 1	0.26	-0.90
7	examine countertop 1	0.17	-0.28
8	go to cabinet 1	0.29	0.31
9	examine cabinet 1	0.35	0.27
10	move soapbottle 1 to cabinet 1	0.26	**0.89**

observations follow the grammar and Footprint conventions of Appendix A. The credit peak falls well before the end of the trajectory, at step 7, the revisit of the processing location where the heating dependency chain converges, four steps ahead of the terminal placement; the processing chain's earlier approach (step 4, $+0.95$) and the heating action itself (step 6, $+0.51$) also carry strong credit, while the terminal placement (step 11, $+0.08$), which merely deposits an already-processed object into a receptacle the agent had explored earlier, stays near neutral. Credit therefore concentrates on the mid-trajectory transformation that decides the outcome, not on the trajectory's last action.

Table 12: Qualitative case study on a complete representative successful Heat-type episode (twelve steps; actions and observations follow the ALFWorld grammar and Footprint conventions of Appendix A). The structural credit peaks at step 7, the processing-location revisit four steps before the terminal placement, with the earlier processing-location approach (step 4) and the heating action (step 6) also well above neutral; the terminal placement (step 11) stays near neutral. Raw teacher-student KL is reported for contrast: its maximum (step 8) coincides with the structural-credit minimum. Bold marks each column's peak.

Step	Action	Raw KL	$\ell_{\text{graph}}(t)$
0	go to fridge 1	0.31	-0.75
1	examine fridge 1	0.44	-1.12
2	go to countertop 1	0.28	-0.68
3	take potato 1 from countertop 1	0.22	-0.82
4	go to microwave 1	0.24	0.95
5	open microwave 1	0.26	-0.04
6	heat potato 1 with microwave 1	0.17	0.51
7	go to microwave 1	0.15	**1.02**
8	go to shelf 1	**0.58**	-1.48
9	go to fridge 1	0.35	-0.66
10	open fridge 1	0.29	-0.86
11	put potato 1 in/on fridge 1	0.41	0.08