WHICH STEPS MATTER? GRAPH-GUIDED SELECTION FOR SINGLE-STEP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for long-horizon agents often requires generating complete interaction trajectories before obtaining a training signal. Repeated reasoning, tool calls, and environment responses make this process expensive, even when the decision of interest occurs much earlier. We propose KStep, a framework that concentrates policy optimization at structurally important decision contexts. From existing trajectories, KStep constructs local state-transition graphs and scores intermediate landmarks by the relative loss of connectivity after their removal. Selected steps become single-step reinforcement learning tasks: the policy receives the full recorded prefix, samples a group of next-turn responses, and updates using group-relative advantages from a reference-action matching reward. Unlike ATLaS, which applies supervised fine-tuning to selected steps, this procedure optimizes newly sampled actions without completing the remaining trajectory. The framework accommodates mathematical reasoning and ALFWorld through domain-specific state adapters. On AIME, KStep exceeds an ATLaS-selection-plus-single-step-RL baseline on two backbones by 3.06 and 1.60 macro-average percentage points. On ALFWorld's seen split, the corresponding success-rate gains are 6.43 and 3.58 points. Full-rollout GRPO achieves higher final-task performance on both domains. The resulting formulation separates where an agent practices from how its sampled decisions are optimized.

1 INTRODUCTION

Language-model agents solve tasks through sequences of reasoning, actions, and observations. A mathematical agent may inspect an expression, write code, read the result, and revise its solution; a household agent may search for an object, acquire it, change its state, and place it at a destination. In both cases, a successful outcome depends on decisions distributed across a long history. Interaction frameworks such as ReAct make these decisions explicit (Yao et al., 2023), but learning from them remains costly when every policy update requires complete new trajectories.

The bottleneck is not only the length of one model response. Full-rollout reinforcement learning (RL) repeatedly generates intermediate responses, waits for tools or environments, and processes the resulting growing context. A terminal correctness signal may arrive many turns after the action that made progress possible. Sampling multiple trajectories per instruction multiplies this cost. For an already recorded decision near the middle of a task, re-generating the preceding trajectory and completing the entire suffix can be an expensive way to obtain another training example at that location. This motivates a different allocation of online generation: reuse the recorded context and practice a consequential next decision directly.

Existing trajectories offer such contexts, but they do not make all steps equally useful. Some responses repeat established plans or execute routine transitions; others establish a state on which downstream progress depends. ATLaS (Chen et al., 2025) demonstrates the value of selecting critical steps for agent tuning. It uses a selector language model to identify planning, observation, action, and self-correction steps, retains the surrounding trajectory as context, and applies supervised fine-tuning (SFT) only to selected targets. This is a natural starting point for deciding *where* to train. We extend this idea from teacher-forced imitation to policy exploration at selected contexts.

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.

We propose KStep, which couples *landmark-guided context selection* with *single-step policy optimization*. Its selector asks a structural question: how much locally supported downstream connectivity remains when an intermediate landmark is removed? Compatible trajectories supply a local graph of abstract states and observed transitions. Comparing connectivity before and after deleting the landmark identifies decision locations whose observed progress depends strongly on that state. Each selected location then supplies a complete historical prefix and a reference next action. The current policy samples several alternative next-turn responses from the prefix, receives a local reference-action matching reward, and is updated with group-relative advantages. No generated suffix is required for this training reward.

This design separates two choices often conflated in trajectory learning: the distribution of contexts on which the agent practices, and the objective used to improve its behavior at those contexts. The graph acts offline on the first choice; the reward and RL optimizer act online on the second. At inference, the agent carries out complete tasks without a reference action or support graph. We evaluate the resulting policy on these full tasks.

Our contributions are a local landmark-dependence formulation, its integration with context-preserving single-step RL, and an evaluation structure that separates selector effects from optimization effects. We provide the graph interpretation and specify the resulting local optimization objective. Experiments on AIME and ALFWorld compare KStep with an ATLaS-selection-plus-single-step-RL adaptation and full-trajectory training baselines. Across both backbones and domains, KStep improves over the adapted selector baseline, while full-rollout GRPO achieves the highest task performance. A trajectory-retention ablation examines how the amount of selected data affects learning.

2 Related Work

Language-model agents combine reasoning, planning, and environment-facing actions to solve interactive tasks. ReAct interleaves reasoning with actions and observations (Yao et al., 2023), while Reflexion uses verbal feedback and memory to revise behavior without gradient-based policy updates (Shinn et al., 2023). These approaches establish useful interaction patterns for sequential decision making. ALFWorld provides a complementary testbed in which success requires a sequence of grounded household actions (Shridhar et al., 2021). Our focus is the training procedure for such agents, with an emphasis on selecting decision contexts for parameter updates.

Agent tuning transfers interaction skills into model parameters. FireAct studies fine-tuning on trajectories from multiple tasks and prompting methods (Chen et al., 2023); AgentTuning mixes high-quality interaction trajectories with general instruction data to improve agent abilities while retaining broader capabilities (Zeng et al., 2023). AgentBank scales diverse trajectory supervision (Song et al., 2024), and AgentOhana standardizes heterogeneous trajectories and their training pipeline (Zhang et al., 2024). These works emphasize the quality, coverage, and organization of agent data. KStep addresses a complementary choice within a trajectory bank: which decision contexts should receive additional policy optimization. DAgger (Ross et al., 2011) studies training on states visited by the learner through iterative data aggregation. This perspective highlights the role of the context distribution: KStep constructs that distribution by selecting intermediate histories from recorded experience.

Interaction-based improvement offers an alternative to static imitation. ReST meets ReAct iteratively trains from agent trajectories with AI feedback (Aksitov et al., 2023), while AgentGym supports exploration across diverse environments and agent self-evolution through AgentEvol (Xi et al., 2024). Such feedback can improve behavior beyond the original demonstrations, but collecting multi-turn interactions retains the cost of generating actions and receiving environment responses. KStep instead samples new actions at recorded intermediate histories and stops after one assistant turn. The resulting exploration compares alternative next responses at a shared decision point, while deployment uses the updated policy for complete multi-turn interaction.

The closest selection method is ATLaS (Chen et al., 2025). Its LLM selector identifies plan creation, critical observation, critical action, and self-correction steps, selecting at most 30% in the main setting. The original method retains the full trajectory context but masks SFT loss outside selected targets. Its value-function comparison additionally examines the expense of repeated rollouts from

intermediate states. We share the motivation to concentrate learning on consequential steps, but separate two changes: graph-based landmark deletion replaces semantic step selection, and single-step RL replaces teacher-forced imitation with optimization of sampled responses. Either selector can supply the same RL interface. Our *ATLaS Selection + Single-Step RL* baseline combines the ATLaS selector with our single-step RL optimizer.

Step-level feedback is also central to credit assignment. Iterative Process Refinement (IPR) estimates step-level rewards through Monte Carlo rollouts and compares generated actions with expert actions to construct contrastive training pairs (Xiong et al., 2024). Process supervision scores intermediate reasoning (Lightman et al., 2023), while RUDDER decomposes delayed returns (Arjona-Medina et al., 2019). GRPO supplies a group-relative policy objective without a separate learned value baseline (Shao et al., 2024). KStep uses that objective with a local reference-matching reward computed directly from candidate responses. Rollout-based value estimation scores an action through its downstream outcomes, whereas structural matching scores its tool names and arguments. The graph score selects contexts, and within-context reward comparisons supply the action advantages used by the optimizer.

Finally, landmarks, temporal abstraction, and graph bottlenecks provide structural perspectives on sequential decisions (Hoffmann et al., 2004; Sutton et al., 1999; Menache et al., 2002). Betweenness centrality characterizes paths through nodes (Freeman, 1977); KStep instead measures the loss of fixed-endpoint path mass after a query-specific landmark deletion, keeping surviving weights unchanged. Routes through the landmark and routes that bypass it jointly determine the score. Influence-based and gradient-alignment methods select data by estimated effects on target learning objectives (Koh & Liang, 2017; Xia et al., 2024). Structural selection answers a different question: where supported progress depends on an intermediate state. These notions of usefulness are complementary and provide distinct criteria for allocating training effort.

3 METHOD

KStep converts recorded long-horizon experience into single-step RL tasks through landmark-guided selection. Figure 1 summarizes the pipeline: build local support graphs, identify important decision locations, and optimize newly sampled responses at their recorded prefixes. Structural importance determines where to train; a local reward determines which sampled response receives reinforcement.

3.1 FROM TRAJECTORIES TO SINGLE-STEP LEARNING

Let a task provide instruction g and initial observation o_0. An agent conditions on history $h_t = (g, o_0, a_0, o_1, \ldots, a_{t-1}, o_t)$ and samples an assistant response $a_t \sim \pi_\theta(\cdot \mid h_t)$. Executing its action produces the next observation. A trajectory $\tau = (g, o_0, a_0, \ldots, a_{T-1}, o_T)$ has terminal outcome $R(\tau)$. Full-trajectory RL seeks to improve

$$J_{\text{task}}(\theta) = \mathbb{E}_{g,\tau\sim\pi_\theta}[R(\tau)]. \tag{1}$$

This expectation includes the histories induced by the current policy and the environment. In a sparse-reward setting, evaluating one sample of the objective requires continuing to a terminal outcome or an episode limit.

We instead assume an existing training collection $\mathcal{D}$ of recorded trajectories, with validity and outcome information. A selector produces indices $\mathcal{C}_\tau$ of training locations. Every selected location defines a task $d_t = (h_t, a_t^*, \mathcal{U}_t)$, where a_t^* is the recorded reference response and $\mathcal{U}_t$ contains the available tool schemas. The prefix includes all earlier assistant and environment turns, including unselected ones, and ends immediately before the reference response. A local reward $r(a, a_t^*; \mathcal{U}_t)$ compares a newly generated response with the separately stored reference. The resulting single-step objective is

$$J_{\text{step}}(\theta) = \mathbb{E}_{d_t\sim\mathcal{D}_\mathcal{C}}\mathbb{E}_{a\sim\pi_\theta(\cdot|h_t,\mathcal{U}_t)}[r(a, a_t^*; \mathcal{U}_t)]. \tag{2}$$

Equation 2 has two levels of sampling: the outer expectation draws a recorded context from $\mathcal{D}_\mathcal{C}$, and the inner expectation draws a response from the current policy. This local objective uses reference-action reward, whereas Equation 1 uses terminal outcomes from policy-induced trajectories. Selection determines the distribution of practice contexts; policy optimization improves the responses sampled at those contexts.

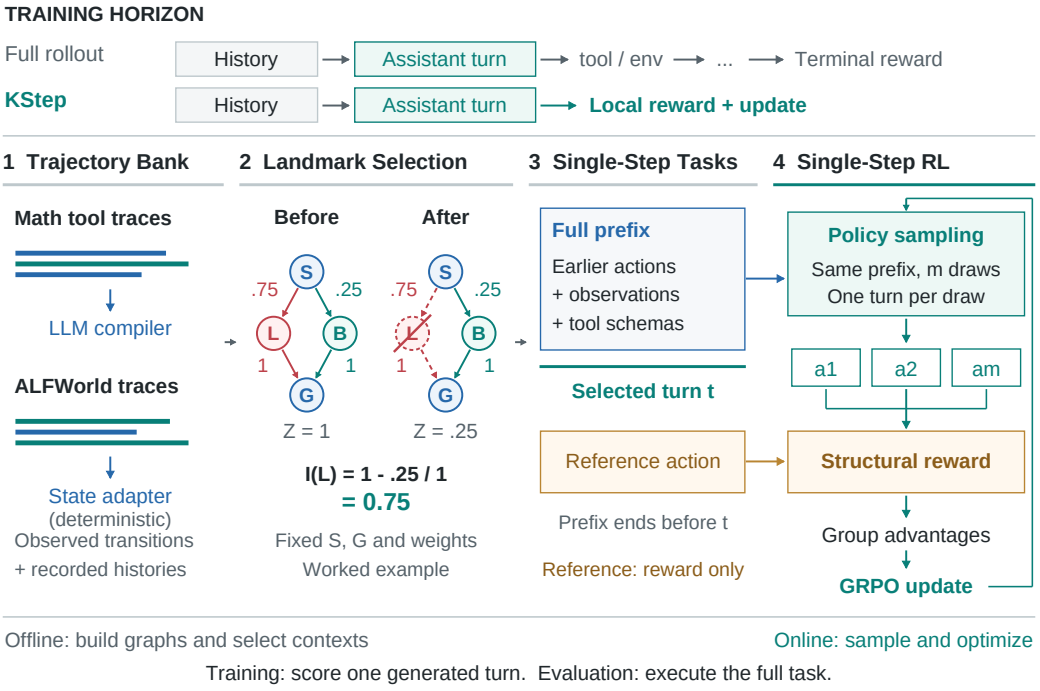

Figure 1: KStep framework. Trajectory states form local graphs, and landmark deletion identifies decision locations for single-step RL. Each selected location supplies a complete history and tool schemas to the policy. Sampled responses and the reference action feed the structural reward, followed by group-relative policy updates. In the worked example, deletion reduces connectivity from 1 to 0.25, giving importance 0.75. Training generates one assistant turn per sample; evaluation executes complete tasks.

Here, a *single step* is one complete assistant turn, which may include reasoning and one or more tool calls. Autoregressive generation continues within that response and stops at its turn boundary or length limit. The training path parses and scores the generated response directly. Deployment and task-level evaluation alternate assistant turns with tool or environment execution until the task terminates.

The selector uses abstract states $x_t = \phi(h_t)$ to describe known facts, progress, and unresolved subgoals. In mathematical tool-use traces, turn boundaries are supplied by the conversation and a compiler may extract relations, operations, and intermediate milestones. For ALFWorld, a deterministic adapter uses recorded observations, inventory changes, action types, and task predicates. These predefined predicates provide the state representation directly. The abstract state supports retrieval and graph construction, while the full history supplies the policy with the detailed observations and actions needed to generate its response.

3.2 Local Graphs and Landmark Dependence

For a recorded query $q = (h_t, a_t^*, h_{t+1})$, retrieve up to k compatible successful transitions from other training tasks. Compatibility requires a shared domain and abstraction schema; retrieval can use recorded state and transition descriptors. Include at most H_s observed suffix transitions from each retrieved position, and deduplicate records that appear in overlapping suffixes. Abstract states form nodes of $G_q = (V_q, E_q)$ and observed transitions form directed edges. Similarity determines which records enter the neighborhood; recorded transitions determine connectivity. The support pool excludes all rollouts of the query task, so the local graph represents routes from other tasks.

Edge weights aggregate nonnegative retrieval relevance, with a per-trajectory cap that balances contributions across source trajectories. Normalizing outgoing weights once defines a reference walk

P_q on G_q; the precise aggregation is given in Appendix C. This reference walk distributes mass over the observed outgoing transitions. Both graph construction and context selection operate on the training trajectory bank.

The query source is $s_q = x_t$. A candidate landmark is a set of equivalent intermediate nodes M_q. A minimal construction uses the observed successor's state class, $M_q = \{v : v \equiv x_{t+1}\}$; a declared semantic predicate can instead group several states. Downstream targets T_q represent a separately specified later achievement or terminal goal, supported by recorded observations. We require

$$s_q \notin T_q, \qquad M_q \cap (\{s_q\} \cup T_q) = \varnothing. \tag{3}$$

These boundaries fix the decision context and downstream achievement while varying an intermediate state. Source-equivalent and terminal-achievement steps receive the unassessed marker $\perp$.

Let $\mathcal{P}_H(s, T; G)$ be walks of at most H edges that terminate on their first visit to T, and let $\mu_q(p) = \prod_{e \in p} P_q(e)$. Connectivity is the bounded first-passage mass

$$Z_q = Z_H(G_q; s_q, T_q) = \sum_{p \in \mathcal{P}_H(s_q, T_q; G_q)} \mu_q(p). \tag{4}$$

Removing a landmark deletes its nodes and incident edges, leaving the endpoints, horizon, and surviving weights unchanged. Define

$$Z_q^{-\ell} = \sum_{\substack{p \in \mathcal{P}_H(s_q, T_q; G_q) \\ p \cap M_q = \varnothing}} \mu_q(p), \qquad \rho_q = \frac{Z_q^{-\ell}}{Z_q}, \qquad I_q = 1 - \rho_q. \tag{5}$$

Thus ρ_q is the retained-connectivity ratio and I_q is importance: larger values mean more original path mass depends on the landmark. Surviving edges retain their original weights. For example, if 0.75 of supported mass passes through a landmark and 0.25 bypasses it, deletion leaves 0.25 and gives $I_q = 0.75$. The same scale before and after deletion makes the ratio directly express the fraction of retained support.

Importance is specific to a source and downstream objective. A frequent state can score low when routes bypass it; a sparsely connected state can score high when most successful routes require it. Examples include acquiring an object before placement and establishing a representation for later computations. The score captures dependence on intermediate progress rather than state frequency or trajectory position.

With these conventions, $I_q \in [0, 1]$ equals the probability that a successful reference walk visits M_q, conditional on hitting T_q within H steps. Paths through the landmark contribute to the removed mass, while bypass paths contribute to the retained mass. We return the unassessed marker $\perp$ when the candidate is absent, the boundaries are invalid, support is insufficient, or Z_q falls below a declared threshold ζ. Two dynamic-programming passes compute the score in $O(H|E_q|)$ time. Formal properties, computation, and worked examples appear in Appendices A–D.

3.3 Turning Selected Steps into RL Tasks

Assign each assessed score to the recorded action entering its candidate state class. The default hard-selection rule keeps the top $\lceil r n_\tau \rceil$ of n_τ assessed steps within a trajectory, with retention fraction r and seeded tie-breaking. Valid unassessed steps are retained for coverage; invalid or unexecuted responses are excluded. The denominator n_τ counts assessed decision locations, and each retained location supplies one training task.

For every retained index, copy all messages before the reference response into h_t and store a_t^* separately. The policy input consists of h_t and the tool schemas; compiler annotations and scoring information remain in task metadata. Earlier unselected actions remain in the prefix along with their observations. This preserves the original decision context at every retained location. The selected tasks can be shuffled and batched independently of the original trajectory order.

The graph score controls which contexts enter $\mathcal{D}_\mathcal{C}$. Within each selected context, the local reward compares newly sampled responses. Thus, structural selection allocates practice across decision locations, and reward-based optimization allocates probability across candidate actions. The score-derived `spci_weight` is stored as diagnostic metadata. Equations 6–8 use the response rewards and group-relative advantages directly.

3.4 Local Reward and Group-Relative Optimization

For each d_t, sample m responses $a^{(1)}, \ldots, a^{(m)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid h_t, \mathcal{U}_t)$. The reward parses their tool calls and compares them with the reference. Agreement about the presence of tool calls yields $r_{\text{presence}} \in \{-1, 1\}$. When both responses contain calls, the structural similarity combines tool-name multiset F1 with an optimal unordered matching of calls, whose argument similarity uses key overlap and token overlap. The configured reward is

$$r_i = r_{\text{presence},i} + \alpha F_{\text{name},i} + (1-\alpha) S_{\text{match},i} - p_{\text{format},i} - p_{\text{language},i}, \qquad \alpha = 0.3. \tag{6}$$

For presence mismatches, both similarity terms are zero; content-only references use a text-similarity branch. Appendix G specifies these cases and the optional format and language deductions.

The reward is computed from the response structure as soon as the assistant turn finishes. Tool-name agreement rewards the choice of operation, argument matching rewards its concrete instantiation, and format constraints enforce the expected response interface. Each sampled action is scored against the same reference before group normalization. Task-level evaluation uses environment success or final-answer correctness.

Let $\bar{r}$ and σ_r be the mean and standard deviation within the group. Group-relative advantages are

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon}, \qquad u_{ij}(\theta) = \frac{\pi_\theta(a_j^{(i)} \mid h_t, \mathcal{U}_t, a_{<j}^{(i)})}{\pi_{\theta_{\text{old}}}(a_j^{(i)} \mid h_t, \mathcal{U}_t, a_{<j}^{(i)})}. \tag{7}$$

The clipped token-level surrogate follows GRPO (Shao et al., 2024):

$$\mathcal{J}(\theta) = \mathbb{E}_{d_t, \{a^{(i)}\}} \left[\frac{1}{m} \sum_{i=1}^{m} \frac{1}{|a^{(i)}|} \sum_j \min(u_{ij} A_i, \text{clip}(u_{ij}, 1-\epsilon_-, 1+\epsilon_+) A_i) \right]. \tag{8}$$

The loss mask covers generated-response tokens, with the prefix serving as conditioning context. Selected-step SFT maximizes reference-token likelihood; the RL update increases or decreases sampled-action probability according to group-relative reward. Responses above the group mean receive positive advantages, and those below it receive negative advantages. The Slime recipe uses $\epsilon_- = 0.2$, $\epsilon_+ = 0.28$, an entropy term of coefficient 0.001, and zero KL coefficient. Zero-variance groups are filtered; optimization uses groups with a reward contrast among their responses.

4 Experiments

We study whether landmark-guided selection improves single-step agent learning, and how this training pipeline compares with full-trajectory SFT and RL. We evaluate mathematical reasoning on AIME and interactive household tasks on ALFWorld.

4.1 Environments

Our mathematical evaluation covers AIME24, AIME25, and AIME26. These competition problems require multi-stage reasoning, often supported by code execution, with correctness assessed from the final answer. We report mean@16 and its macro average across the three years. Each evaluation trajectory solves a complete problem through successive reasoning and tool-use decisions.

ALFWorld is a text-based environment with six families of household tasks, including object placement, examination under light, and placement after cleaning, heating, or cooling (Shridhar et al., 2021). Tasks require exploration and dependent actions across multiple observations. Its explicit action schema and observable state predicates support deterministic state extraction, complementing the text compiler used for mathematical trajectories. We evaluate on the 140 tasks in ALFWorld's seen split and report overall success across all six task families in a single All column.

4.2 Implementation Details

The mathematical comparison uses Qwen2.5-7B and Qwen3-8B (Qwen et al., 2024; Yang et al., 2025). For each selected training location, the policy receives the original history and tool definitions, while the reference response is stored separately for reward computation. The optimization

Table 1: Main results on mathematical reasoning and ALFWorld (%; higher is better). AIME reports mean@16 scores; Avg. is the mean across the three AIME years. ALFWorld (Seen) reports overall success on 140 tasks across all six task families (All). Our ATLaS adaptation combines the original selector with single-step RL. Bold marks the best score per column within each backbone.

	AIME mean@16				ALFWorld (Seen)
Method	2024	2025	2026	Avg.	All
Qwen2.5-7B					
Base	10.80	9.80	8.90	9.83	35.71
SFT	12.70	10.90	11.40	11.67	55.71
GRPO	**37.90**	**30.80**	**27.90**	**32.20**	**75.00**
Single-Step RL					
ATLaS Selection + Single-Step RL	27.50	23.33	23.13	24.65	64.28
KStep	30.42	26.04	26.67	27.71	70.71
Qwen3-8B					
Base	52.50	41.40	44.70	46.20	47.85
SFT	54.80	46.88	52.71	51.46	70.71
GRPO	**76.25**	**64.79**	**68.95**	**70.00**	**83.57**
Single-Step RL					
ATLaS Selection + Single-Step RL	70.60	57.00	64.10	63.90	71.42
KStep	73.90	57.80	64.79	65.50	75.00

procedure samples a group of next-turn responses and applies GRPO using the structural reward in Section 3. Each sample ends at the assistant-turn boundary and is scored directly from its content and tool calls. Reference responses are consumed by the reward module, and full histories condition policy generation.

The Slime recipe uses 128 prompts per rollout batch, 16 responses per prompt, and a learning rate of 10^{-6}. The response and context limits are 4,096 and 32,768 tokens.

4.3 Baselines

We organize the comparisons around the training objective and the choice of training locations. Base denotes the starting-model reference. SFT uses supervised trajectory learning, while GRPO is the full-rollout RL reference. The Single-Step RL group in Table 1 contains ATLaS Selection + Single-Step RL and KStep. ATLaS Selection + Single-Step RL uses the semantic critical-step selector of Chen et al. (2025) with single-step policy optimization. KStep replaces this selector with local landmark-deletion scoring. The two single-step methods compare structural and semantic selection under the same form of policy optimization. SFT and full-rollout GRPO provide complementary references for supervised imitation and terminal-reward optimization.

Relevant ALFWorld baselines include prompting-based agents ReAct and Reflexion (Yao et al., 2023; Shinn et al., 2023), trajectory-trained AgentLM and AgentEvol (Zeng et al., 2023; Xi et al., 2024), and the full, random, perplexity, non-critical, and critical-step SFT variants reported by ATLaS (Chen et al., 2025). Together, these approaches span inference-time reasoning, trajectory supervision, and selective step learning. Our comparison focuses on selection within the single-step RL setting using the two Qwen backbones.

4.4 Main Results

Table 1 groups AIME and ALFWorld under the same backbone and training-method rows. ALFWorld (Seen) reports overall task success, while the AIME average covers the three competition years. On AIME, KStep exceeds ATLaS Selection + Single-Step RL on all six backbone–benchmark pairs. For Qwen2.5-7B, the gains are 2.92, 2.71, and 3.54 percentage points on AIME24, AIME25, and AIME26, respectively, increasing the macro average from 24.65 to 27.71. For Qwen3-8B, the gains are 3.30, 0.80, and 0.69 points, with the macro average rising from 63.90 to 65.50. The corresponding mean improvements are 3.06 and 1.60 points.

Relative to SFT, KStep improves the macro average by 16.04 points on Qwen2.5-7B and 14.03 points on Qwen3-8B. Full-rollout GRPO achieves the highest score on every AIME benchmark for both backbones. KStep trails GRPO by 4.49 points on average for Qwen2.5-7B and 4.50 points for Qwen3-8B. On Qwen3-8B, the GRPO scores are 76.25, 64.79, and 68.95, with a macro average of 70.00, compared with 65.50 for KStep.

On ALFWorld's seen split, the All success rates for KStep are 70.71% on Qwen2.5-7B and 75.00% on Qwen3-8B, compared with 64.28% and 71.42% for ATLaS Selection + Single-Step RL, giving gains of 6.43 and 3.58 percentage points, respectively. KStep also exceeds SFT (55.71% and 70.71%) but remains below full-rollout GRPO (75.00% and 83.57%) on both backbones.

4.5 Ablation Study

We vary the fraction of ranked source trajectories retained for single-step learning. Table 2 reports Qwen2.5-7B and Qwen3-8B results at 10%, 30%, 50%, and 100% retention. The 30% settings reproduce the KStep results in Table 1. ALFWorld reports All on the same 140-task seen split as the main table.

For Qwen2.5-7B, Top 30% achieves the highest score on all three AIME benchmarks in Table 2, with a macro average of 27.71. The corresponding averages for Top 10%, Top 50%, and 100% are 23.47, 25.90, and 24.58, giving gains of 4.24, 1.81, and 3.13 percentage points.

For Qwen3-8B, Top 30% also has the highest macro average, 65.50, compared with 52.43, 64.02, and 60.28 at 10%, 50%, and 100% retention. Its mean gains are 13.07, 1.47, and 5.22 percentage points, respectively. The per-benchmark optimum varies: Top 50% achieves 74.16 on AIME24, exceeding Top 30% by 0.26 points, whereas Top 30% remains best on AIME25 and AIME26. Thus both backbones show non-monotonic macro-average performance as retention increases, but the best fraction can differ across benchmarks. This pattern suggests a trade-off between coverage and training focus.

ALFWorld further shows that the best retention fraction depends on the backbone and environment. For Qwen2.5-7B, the scores at 10%, 30%, 50%, and 100% are 54.28%, 70.71%, 74.28%, and 68.57%, respectively. Top 50% is best, exceeding Top 30% by 3.57 points and retaining all eligible trajectories by 5.71 points. For Qwen3-8B, the corresponding scores are 62.14%, 75.00%, 70.71%, and 72.85%. Top 30% is best, exceeding Top 50% by 4.29 points and 100% by 2.15 points. Both backbones exhibit non-monotonic performance as the retained fraction increases. Table 1 uses the common Top 30% setting across model–benchmark pairs.

Intermediate retention performs best across both domains. Top 10% and 100% fall below Top 30% on every backbone–benchmark pair. Increasing retention from 10% to 30% improves all eight scores, while further expansion changes the balance between selection and coverage. Together, these results favor training on structurally important decisions drawn from a sufficiently broad range of contexts.

Ranking the four retention settings within each backbone–benchmark pair summarizes their cross-domain consistency. Top 30% ranks first in six of the eight comparisons and second in the other two, giving an average rank of 1.25. Top 50% has an average rank of 1.88, followed by 100% at 2.88 and Top 10% at 4.00. The two wins for Top 50% occur on Qwen3-8B AIME24 and Qwen2.5-7B ALFWorld. This ranking complements the domain-specific scores: intermediate retention leads across all eight comparisons, and Top 30% consistently places among the top two. It therefore provides a strong common setting across the mathematical and interactive tasks studied here.

Trajectory retention and step retention are different controls. Here, p specifies which complete source trajectories enter task construction, whereas the method's r selects assessed decision locations within each retained trajectory. The 100% row uses all eligible source trajectories while retaining the within-trajectory step-selection rule and single-step RL objective. The source trajectory supplies the full sequence of observations, while the selected locations determine which next-turn decisions receive policy updates. This two-level organization connects trajectory diversity with targeted practice within each trajectory.

Table 2: Trajectory-retention ablation. We retain the top $p\%$ of ranked source trajectories before extracting single-step tasks; 100% retains all eligible trajectories. ALFWorld reports All over the 140-task seen split. Top 30% reproduces the KStep rows in Table 1. Bold marks the best score per benchmark within each backbone. Here p controls source trajectories and r controls steps within each trajectory.

	AIME mean@16			ALFWorld (Seen)
Trajectory retention	2024	2025	2026	All
Qwen2.5-7B				
Top 10%	24.58	22.70	23.12	54.28
Top 30%	**30.42**	**26.04**	**26.67**	70.71
Top 50%	27.91	24.58	25.20	**74.28**
100% (all eligible)	26.87	23.33	23.54	68.57
Qwen3-8B				
Top 10%	57.70	49.38	50.20	62.14
Top 30%	73.90	**57.80**	**64.79**	**75.00**
Top 50%	**74.16**	55.41	62.50	70.71
100% (all eligible)	67.08	53.75	60.00	72.85

5 LIMITATIONS

KStep learns from a fixed bank of decision contexts. As the policy changes, its visited histories can shift beyond that bank, motivating periodic collection and reselection of trajectories. Graph scores depend on the state abstraction and the alternatives represented in retrieved support: missing bypass routes increase the measured dependence on a landmark. Richer support and state representations are therefore important directions for extending the selection procedure.

The local reward measures reference-action structure. Semantically equivalent actions can have different surface forms, while structurally similar actions can produce different outcomes. Execution-based or semantic rewards offer extensions of the same selected-context interface. The ALFWorld evaluation uses 140 seen tasks; extending evaluation to unseen environments would examine transfer across new layouts and objects. A broader evaluation can also study variation across seeds and the balance among trajectory collection, graph construction, prefix processing, and online generation.

6 CONCLUSION

KStep combines landmark-guided context selection with single-step reinforcement learning. Local graphs identify decision locations by the loss of connectivity after landmark deletion. At each selected prefix, the policy samples new assistant turns and learns from reference-matching rewards and group-relative advantages. Training stops after one turn, while evaluation retains complete-task execution.

On AIME, KStep exceeds ATLaS Selection + Single-Step RL by 3.06 and 1.60 macro-average percentage points on the two backbones, while trailing full-rollout GRPO by 4.49 and 4.50 points. On ALFWorld's seen split, the gains over the adapted selector are 6.43 and 3.58 points; GRPO also achieves the highest scores. Top 30% retention gives the best AIME macro average on both backbones, while Top 50% is best on Qwen3-8B AIME24. ALFWorld favors Top 50% for Qwen2.5-7B and Top 30% for Qwen3-8B, showing that the best fraction varies across models and tasks.

Future work will examine end-to-end training cost, semantic rewards, and context selection beyond the recorded trajectory bank. KStep provides a framework for studying these questions by separating the selection of training contexts from the optimization of sampled decisions. More broadly, long-horizon reasoning and short-horizon optimization can coexist: the complete history supplies the information needed for a decision, while the learning objective focuses on the next response. This distinction connects trajectory-level experience to targeted policy improvement.

REFERENCES

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, Manzil Zaheer, Felix Yu, and Sanjiv Kumar. ReST meets ReAct: Self-improvement for multi-step reasoning LLM agent. arXiv preprint arXiv:2312.10003, 2023. URL https://arxiv.org/abs/2312.10003.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. arXiv preprint arXiv:1806.07857, revised 2019, 2019. URL https://arxiv.org/abs/1806.07857.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. FireAct: Toward language agent fine-tuning. arXiv preprint arXiv:2310.05915, 2023. URL https://arxiv.org/abs/2310.05915.

Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. ATLaS: Agent tuning via learning critical steps. arXiv preprint arXiv:2503.02197v1, 2025. URL https://arxiv.org/abs/2503.02197v1.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. doi: 10.2307/3033543. URL https://doi.org/10.2307/3033543.

J. Hoffmann, J. Porteous, and L. Sebastia. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 22:215–278, 2004. doi: 10.1613/jair.1492. URL https://doi.org/10.1613/jair.1492.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023. URL https://arxiv.org/abs/2305.20050.

Ishai Menache, Shie Mannor, and Nahum Shimkin. Q-Cut: Dynamic discovery of sub-goals in reinforcement learning. In *Machine Learning: ECML 2002*, pp. 295–306. Springer, 2002. doi: 10.1007/3-540-36755-1_25. URL https://doi.org/10.1007/3-540-36755-1_25.

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. arXiv preprint arXiv:2303.11366, 2023. URL https://arxiv.org/abs/2303.11366.

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

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. AgentBank: Towards generalized LLM agents via fine-tuning on 50000+ interaction trajectories. arXiv preprint arXiv:2410.07706, 2024. URL https://arxiv.org/abs/2410.07706.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evolving large language model-based agents across diverse environments. arXiv preprint arXiv:2406.04151, 2024. URL https://arxiv.org/abs/2406.04151.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. arXiv preprint arXiv:2402.04333, 2024. URL https://arxiv.org/abs/2402.04333.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! LLM agent learning via iterative step-level process refinement. arXiv preprint arXiv:2406.11176, 2024. URL https://arxiv.org/abs/2406.11176.

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

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. arXiv preprint arXiv:2310.12823, 2023. URL https://arxiv.org/abs/2310.12823.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Ming Zhu, Juntao Tan, Thai Hoang, Zuxin Liu, Liangwei Yang, Yihao Feng, Shirley Kokane, Tulika Awalgaonkar, Juan Carlos Niebles, Silvio Savarese, Shelby Heinecke, Huan Wang, and Caiming Xiong. AgentOhana: Design unified data and training pipeline for effective agent learning. arXiv preprint arXiv:2402.15506, 2024. URL https://arxiv.org/abs/2402.15506.

A Properties and Interpretation

The following statements assume nonnegative fixed reference weights, bounded first-passage walks, and disjoint boundaries as in Equation 3. Proofs appear in Appendix B.

Proposition 1 (Conditional path dependence). *For $Z_q > 0$, the landmark importance equals the reference-walk probability of visiting M_q before first reaching T_q, conditional on reaching T_q within H steps. Consequently $0 \le I_q \le 1$. Moreover, $I_q = 1$ exactly when every positive-mass first-passage walk of length at most H intersects M_q, and $I_q = 0$ exactly when none does.*

The proposition makes the meaning of "important" precise: a large fraction of locally supported successful path mass passes through the landmark. The conditioning event is arrival at the fixed target set within the walk horizon, and the probability measure is defined by the reference weights on the local graph.

Proposition 2 (Deletion-set monotonicity). *For fixed G_q, s_q, T_q, H, P_q and deletion sets $M_1 \subseteq M_2$ satisfying Equation 3, $I(M_1) \le I(M_2)$ whenever $Z_q > 0$.*

Larger deletion classes remove at least as much successful mass as their subsets. An exact intermediate state and a semantic predicate spanning several states therefore induce different ranking resolutions. If every supported successful path satisfies a predicate, deleting its entire realization set gives importance one. All such necessary predicates share this maximal score.

Proposition 3 (Sensitivity of the ratio). *Consider two graph estimates for the same query, landmark, endpoints, and horizon. Let their before/after masses be (Z, A) and $(\widetilde{Z}, \widetilde{A})$, with $0 \le A \le Z$, $0 \le \widetilde{A} \le \widetilde{Z}$, and $Z, \widetilde{Z} \ge \zeta > 0$. If both mass estimates differ by at most δ, then $|I - \widetilde{I}| \le \min\{1, 2\delta/\zeta\}$.*

The bound relates score sensitivity to the pre-deletion mass. For a fixed mass perturbation δ, a larger lower bound ζ gives a tighter bound on the score change. The scoring rule records Z_q and returns $\perp$ below this threshold. Increasing retrieval size or horizon can add both bypass routes and routes through the landmark, so either an increase or decrease in importance is possible.

Structural dependence versus action dependence. Suppose two actions lead from the same source into the same landmark. Deleting one action can leave the other route intact; deleting the landmark removes both. Node deletion answers whether supported progress depends on a state class. Edge deletion answers whether it depends on a transition. The two operators thus distinguish a shared intermediate achievement from an individual action that reaches it.

B Proofs

Proof of Proposition 1. Partition the finite set of first-passage walks into $\mathcal{P}^{\mathrm{hit}} = \{p : p \cap M_q \neq \varnothing\}$ and $\mathcal{P}^{\mathrm{avoid}} = \{p : p \cap M_q = \varnothing\}$. Deletion removes precisely the former set and leaves every surviving $\mu_q(p)$ unchanged. Thus

$$Z_q - Z_q^{-\ell} = \sum_{p \in \mathcal{P}^{\mathrm{hit}}} \mu_q(p), \qquad I_q = \frac{\sum_{p \in \mathcal{P}^{\mathrm{hit}}} \mu_q(p)}{\sum_{p \in \mathcal{P}^{\mathrm{hit}} \cup \mathcal{P}^{\mathrm{avoid}}} \mu_q(p)}.$$

The ratio is the conditional visitation probability. Nonnegativity and containment of the hit set imply boundedness. A ratio of one means that all avoiding walks have zero mass; a ratio of zero means that all hitting walks have zero mass. The first-passage convention makes the terminated walk events disjoint, so $Z_q \le 1$ even when non-target nodes contain cycles. □

Proof of Proposition 2. Every walk avoiding M_2 also avoids M_1. Since all walk weights are nonnegative and unchanged, $Z^{-M_2} \le Z^{-M_1}$. Subtraction from one after division by the same positive Z_q gives the result. □

Proof of Proposition 3. Using $\widetilde{A}/\widetilde{Z} \le 1$ gives

$$|I - \widetilde{I}| = \left|\frac{A}{Z} - \frac{\widetilde{A}}{\widetilde{Z}}\right| \le \frac{|A - \widetilde{A}|}{Z} + \frac{\widetilde{A}\,|Z - \widetilde{Z}|}{Z\widetilde{Z}} \le \frac{2\delta}{\zeta}.$$

Both scores belong to $[0, 1]$, which also bounds their difference by one. □

C Scoring Algorithm

For an edge e, let $\mathcal{R}_e$ be its distinct supporting records and $r(q, j) \in [0, 1]$ their retrieval relevance. The aggregation caps evidence within each support trajectory:

$$b_q(e) = \sum_{\tau \in \mathcal{D}^+_{-q}} \max\big(\{r(q, j) : j \in \tau \cap \mathcal{R}_e\} \cup \{0\}\big), \qquad P_q(e) = \frac{b_q(e)}{\sum_{e' : \mathrm{src}(e') = \mathrm{src}(e)} b_q(e')}. \quad (9)$$

Here $\mathcal{D}^+_{-q}$ excludes the query task and includes only valid successful support. A zero denominator produces no outgoing mass. The scoring parameters are k, H_s, H, ζ and the minimum distinct support-task count B.

Algorithm 1 KStep offline landmark scoring

Require: Query q, training support $\mathcal{D}$, adapter ϕ, retrieval size k
Require: Suffix horizon H_s, walk horizon H, threshold ζ, support count B
Retrieve compatible successful transitions, excluding the query task
Include observed suffixes up to H_s; deduplicate support records
Build G_q and normalize P_q once using Equation 9
Set source s_q, candidate nodes M_q, and fixed downstream targets T_q
if candidate absent, boundaries invalid, or candidate support tasks $< B$ **then**
    return $\perp$ with support diagnostics
end if
$Z_q \leftarrow \textsc{Mass}(G_q, P_q, s_q, T_q, H, \varnothing)$
if $Z_q < \zeta$ **then**
    return $\perp$ with Z_q and support diagnostics
end if
$Z_q^{-\ell} \leftarrow \textsc{Mass}(G_q, P_q, s_q, T_q, H, M_q)$
return $1 - Z_q^{-\ell}/Z_q$ with both masses and support diagnostics
function $\textsc{Mass}(G, P, s, T, H, M)$
    $f \leftarrow \{s : 1\}$; $z \leftarrow 0$
    for $d = 1, \ldots, H$ **do**
        Initialize an empty zero-valued mass map f'
        for each edge $e = (u, v)$ with $u \notin M$, $v \notin M$, and $f(u) > 0$ **do**
            $a \leftarrow f(u)P(e)$
            if $v \in T$ **then**
                $z \leftarrow z + a$
            else
                $f'(v) \leftarrow f'(v) + a$
            end if
        end for
        $f \leftarrow f'$
    end for
    return z
end function

For completeness, write $P(u, v)$ as the sum of parallel-edge probabilities from u to v. The recurrence for a deletion set M is

$$f_{d+1}^{M}(v) = \mathbf{1}[v \notin T \cup M] \sum_{u \notin T \cup M} f_{d}^{M}(u) P(u, v), \tag{10}$$

$$z_{d+1}^{M} = z_{d}^{M} + \sum_{u \notin T \cup M} \sum_{v \in T} f_{d}^{M}(u) P(u, v), \tag{11}$$

with $f_{0}^{M}(s) = 1$ and $z_{0}^{M} = 0$. These formulas assume $M \cap (\{s\} \cup T) = \varnothing$. Then $z_{H}^{\varnothing} = Z_q$ and $z_{H}^{M} = Z_{q}^{-\ell}$. A bounded horizon ensures termination even when the graph contains cycles; no absorbing-chain inverse is required.

D Analytical Examples and Boundary Cases

A bypassable landmark. Let the edges be $s \to m$, $s \to b$, $m \to c$, $b \to c$, and $c \to g$. Assign probabilities 0.75 and 0.25 to the two outgoing edges of s and probability one to all other edges. For $H = 3$, $T = \{g\}$, and $M = \{m\}$, $Z = 1$, $Z^{-M} = 0.25$, and $I(M) = 0.75$. For $M = \{c\}$, the retained mass is zero and $I(M) = 1$.

Fixed and re-normalized weights. In the preceding graph, deleting m and re-normalizing the source's remaining edge changes $P(s, b)$ from 0.25 to one. The new walk reaches g with probability one, producing a zero relative drop. That quantity measures the new walk's reachability, whereas the fixed-weight value measures the share of the original successful mass that used m.

First-passage target handling. Let $s \to g$ and $g \to g$ both have probability one. Summing visits to g over a horizon H gives a visit count of H. First-passage scoring gives $Z = 1$ for every $H \geq 1$. Each successful walk contributes once to the probability in Proposition 1.

Cycles before success. If $s \to s$ has probability 0.4 and $s \to g$ has probability 0.6, then $Z_H = 1 - 0.4^{H}$. This finite-horizon value includes repeated non-target states without counting the same completed walk more than once.

Absent paths and unsupported candidates. When no supported path reaches T, the denominator is zero and the score is $\perp$. An absent candidate receives the same marker. In contrast, $I = 0$ is assigned to an assessed candidate when the positive successful mass is entirely preserved after deletion. These outcomes distinguish unassessed locations from supported bypasses.

Necessary predicates and score saturation. If every successful supported path must make a predicate true, and M removes all intermediate states satisfying that predicate, then $I(M) = 1$. Repeating this construction for several necessary predicates yields identical scores. Refining the state classes measures dependence on more specific intermediate states. Candidate granularity therefore determines the resolution at which the graph distinguishes decision locations.

E Data and Adapter Contract

The transition schema contains the task identifier, trajectory identifier, time index, observed prefix, executed action, successor observation, abstract source and successor, candidate identifier, validity flag, and terminal outcome. Supporting record identifiers associate graph edges with their source observations. Graph support consists of executed transitions and their observations.

For a structured environment such as ALFWorld, the adapter distinguishes entity instances within a trajectory before applying cross-task role normalization. Observed actions, inventory changes, and object attributes specify milestone achievement. Candidate-state equivalence determines which node set is removed, and a separate target predicate specifies the downstream achievement. Unobserved attributes remain unspecified in the state representation.

In an unstructured environment, a language-model compiler extracts the corresponding fields from the recorded text. Field validation checks the schema, grounding in the source passage, and con-

sistency across adjacent transitions. The retrieved records supply the alternative routes used for the local connectivity comparison.

The mathematical schema additionally contains the problem identifier, original solution text, segment boundaries, token offsets, and answer-verifier result. Compiler annotations are stored as selection metadata, while original messages form the policy context. Problem identifiers define the grouping unit for split assignment: rollouts of one problem belong to the same group. The data contract assigns benchmark problems and their paraphrases to evaluation groups, separate from training and graph support.

F Scoring Variants

Landmark-node deletion, transition-edge deletion, and action-class deletion are separate operators. Similarly, fixed first-passage mass, re-normalized first-passage mass, and repeated-visit mass are separate connectivity definitions. Section 3 specifies landmark-node deletion with fixed first-passage mass. This combination measures the fraction of original successful path mass that passes through the deleted state class.

Action-class deletion instead groups transitions by action semantics. Re-normalization redistributes mass among surviving edges, while repeated-visit scoring accumulates each target visit. Multiplying a score by a confidence factor further changes the ranking. These choices define distinct scoring variants with different interpretations.

G Single-Step Training Interface and Reward

The data interface has four fields: `messages` holds the complete prefix; `label` holds the reference assistant content and tool calls; `tools` holds available tool schemas; and `metadata` preserves trajectory ID, turn index, data source, and score diagnostics. The prompt is formed from `messages` and `tools`; the reward consumes `label`. Task extraction removes obsolete loss-mask fields from prefix messages while retaining their content. The prefix ends at the observation immediately preceding the selected assistant response.

The active reward, `hybrid_unordered_score`, is configured by `KSTEP_REWARD_SCORE_KEY` with mixture parameter `HYBRID_ALPHA=0.3`. For normalized tool calls, let N^* and N be the multisets of reference and predicted names. Name similarity is $F = 2|N^* \cap N|/(|N^*| + |N|)$, with multiset intersection. For a same-name pair of calls with argument keys K^*, K, define

$$s(c^*, c) = \frac{|K^* \cap K|}{|K^* \cup K|} \frac{1}{|K^* \cap K|} \sum_{k \in K^* \cap K} \mathrm{IoU}_{\mathrm{tokens}}(c^*[k], c[k]). \tag{12}$$

The score is zero for differing tool names or no shared argument keys. Non-string values are serialized to text before token comparison. An unordered best assignment maximizes the sum of pairwise scores and divides by the larger of the reference and predicted call counts to form S_{match}; unmatched calls contribute no matching credit. The matching procedure compares the parsed call structure and arguments.

Token overlap uses the checkpoint tokenizer without special tokens. For token counts c^*_v, c_v, multiset IoU is $\sum_v \min(c^*_v, c_v)/\sum_v \max(c^*_v, c_v)$. Two empty content strings score one, and exactly one empty string scores zero. Non-string argument values use sorted-key JSON serialization when possible. Predicted tool presence is detected from parsed calls or a raw tool-call payload; a malformed payload can therefore trigger presence agreement without obtaining name or argument credit. If the reward function raises an exception, the default wrapper returns -2 with an `rm_fallback` diagnostic, unless configured to re-raise the error. With `REWARD_RAISE_ON_ERROR=1`, the launcher re-raises the exception instead of using this fallback.

For the configured hybrid score, the complete branching is

$$r = \begin{cases} -1 - p, & \text{tool-call presence disagrees,} \\ 1 + \alpha F + (1-\alpha) S_{\mathrm{match}} - p, & \text{both contain tool calls,} \\ 1 + \alpha + (1-\alpha)\,\mathrm{IoU}_{\mathrm{tokens}}(a^*, a) - p, & \text{both are content-only,} \end{cases} \tag{13}$$

where $p = p_{\text{format}} + p_{\text{language}}$. With the default nonnegative constraints, the hybrid range is $[-3.1, 2]$. In a content-only group, constant offsets cancel under group centering, leaving relative token similarity to determine the advantages.

The default format deduction is 0.1 when the response lacks a complete `<think>` block. The default language deduction is 2.0 on detected Han, Hiragana, Katakana, Hangul, or Cyrillic script; an optional language identifier can confirm the detection, with regex-only fallback if no model is loaded. These English-language format settings apply to generated response text. Mathematical evaluation uses a final-answer checker, and ALFWorld evaluation uses environment task success.

Algorithm 2 Selected-context single-step policy optimization

Require: Recorded trajectories, selection indices, policy π_θ, group size m

1: Extract $(h_t, a_t^*, \mathcal{U}_t)$ for each selected valid turn
2: **for** each training batch of selected contexts **do**
3: Sample m assistant responses per context from the rollout policy
4: Stop each sample at its assistant-turn boundary or response limit
5: Score response content and tool-call structure against a_t^*
6: Filter zero-variance groups and record sampling counts
7: Compute group-relative advantages from the local rewards
8: Update θ using the clipped generated-token objective
9: Synchronize the rollout policy according to the training schedule
10: **end for**
11: Evaluate the trained policy through complete task execution

DISCLOSURE OF AI USE

Generative AI tools were used primarily to assist with drafting, language polishing, and organization of the manuscript.The authors are responsible for verifying all AI-assisted content and take full responsibility for the final manuscript, including its claims, references, and reported results.