STATE-GRAPH POLICY OPTIMIZATION FOR LONG-HORIZON AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative reinforcement learning removes the need for a learned critic, but trajectory-level advantages provide coarse credit assignment for long-horizon language-model agents. Stepwise group methods refine this signal by comparing actions from a shared anchor state, yet they still score each action using the Monte Carlo return of its own sampled suffix. For long remaining horizons, these suffix returns can accumulate substantial downstream policy and environment stochasticity and cannot exploit future evidence from other trajectories that revisit the same successor states. We introduce State-Graph Policy Optimization (SGPO), a critic-free step-return estimator that shares such evidence across task-matched rollouts. For each rollout group, SGPO merges observed transitions into an empirical state graph, performs certainty-equivalent policy evaluation on the graph, and uses the resulting values to bootstrap a TD(λ) return along each real trajectory. SGPO shares value information only through actually visited states, requires no additional environment interactions or model evaluations, and leaves the outer group-relative optimizer unchanged. We prove that the empirical Bellman system is well posed for $\gamma < 1$, characterize its behavior-policy value target under a Markov anchor abstraction, and show that $\lambda = 1$ recovers the matched Monte Carlo step advantage. Across two Qwen model scales on ALFWorld and WebShop, SGPO achieves higher mean aggregate performance than the strongest evaluated group-based baselines. On WebShop, shuffling successor connectivity substantially reduces performance, providing evidence that the gain depends on the observed transition structure rather than bootstrapping alone.The code is available at https://anonymous.4open.science/r/verl-agent-A5EE.

1 INTRODUCTION

Large language models(Vaswani et al., 2017; Hurst et al., 2024) increasingly act through multi-step interactions (Yao et al., 2022b; Wei et al., 2022) rather than producing a single response. In embodied environments such as ALFWorld (Shridhar et al., 2020) and web-navigation environments such as WebShop (Yao et al., 2022a), WebArena (Zhou et al., 2024a), and Mind2Web (Deng et al., 2023), an agent may make tens of decisions before receiving a sparse success signal. Reinforcement learning (RL) must therefore determine not only whether an episode succeeded, but which actions made progress and which caused failure.

Group-relative algorithms offer a practical alternative to actor–critic training for large models (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023). Given several rollouts of the same task, they form a baseline from the rewards of the group and optimize the policy without a learned value network (Williams, 1992; Sutton et al., 1999; Ahmadian et al., 2024; Shao et al., 2024). This simplicity is attractive, but a trajectory-level advantage assigns the same outcome signal to every action in an episode. GIGPO addresses this limitation by grouping rows that share an anchor state and normalizing their returns locally (Feng et al., 2026). It preserves the critic-free training pipeline while adding a step-level comparison.

The remaining limitation is statistical. At a row i, GIGPO (Feng et al., 2026) uses the complete Monte Carlo return along the sampled suffix. Two actions from the same anchor state are therefore compared using all subsequent policy and environment randomness. The variance grows with the remaining horizon. Moreover, suppose a trajectory reaches a state u after the action being evaluated

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

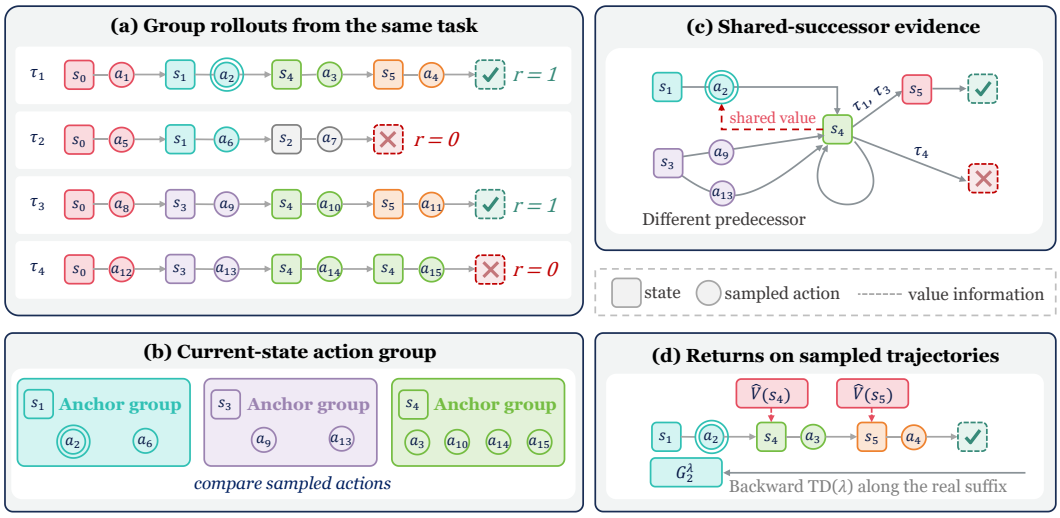


Figure 1: The intuition behind SGPO. (a) A group of rollouts $\tau_1, \ldots, \tau_4$ on the same task, with their success or failure outcomes. (b) Actions that share the current anchor state form an anchor group (s_1, s_3, s_4), within which the sampled actions are compared. (c) Shared-successor evidence: trajectories with different predecessors (through s_1 and s_3) reach the same successor s_4, pooling future value at a node that was actually visited. (d) Along each sampled trajectory, a backward TD(λ) recursion that uses $\widehat{V}$ over the real suffix produces the step return G^{λ}. Boxes denote states, circles denote sampled actions, and dashed arrows carry value information.

and another trajectory reaches the same u from elsewhere. The second trajectory contains relevant evidence about the future of u, but an anchor-local Monte Carlo comparison cannot use it unless both trajectories also share the original predecessor.

We ask whether the already collected rollout group can supply a better bootstrap without giving up the computational advantages of group-relative RL. Our key observation is that rollouts conditioned on the same task and environment instance are samples from the same dynamics. Their observed transitions can be merged into a small empirical state graph. Solving the Bellman equations of the graph yields a certainty-equivalent (CE) value: the expected discounted return under the transition frequencies observed in the group. This value is not a learned critic and is discarded after the current group is processed.

Based on this observation, we propose SGPO. The outer training loop is unchanged: the same rollouts, episode advantage, anchor-state groups, advantage combination, and clipped policy objective are used. SGPO replaces only the step-return estimator. It first solves for the CE value on the group graph, then combines that value with real trajectory rewards through TD(λ). The interpolation parameter makes the bias–variance trade-off explicit: $\lambda = 0$ uses a one-step graph bootstrap, while $\lambda = 1$ ignores the graph value and exactly recovers the Monte Carlo endpoint.

Our contributions are:

- **Transition-aware, critic-free step estimator.** We introduce a transition-aware, critic-free value estimator that pools future evidence at shared successor states inside a rollout group, and integrate it with TD(λ) as a drop-in step-return estimator.
- **Theoretical characterization.** We prove well-posedness of the empirical Bellman system for $\gamma < 1$, identify the behavior-value target under a Markov state abstraction, and establish exact recovery of the single-trajectory Monte Carlo step advantage at $\lambda = 1$.
- **Empirical evidence for both performance and mechanism.** Across two model scales on ALFWorld and WebShop, SGPO improves aggregate agent performance over the strongest evaluated group-based baselines; trace ablations and a shuffled-connectivity intervention further show that intermediate graph bootstrapping and the observed transition structure contribute materially to the gain.

2 Related Work

Group-relative optimization. PPO uses a learned value function and a clipped surrogate objective (Schulman et al., 2017). Reinforce leave-one-out and related group-relative methods replace the critic with a sample baseline (Kool et al., 2019; Ahmadian et al., 2024). GRPO popularized normalized group rewards for LLM reasoning (Shao et al., 2024), and subsequent large-scale systems refine this recipe for reasoning models (Guo et al., 2025; Yu et al., 2026; Liu et al., 2025). For long-horizon agents, GiGPO adds an episode-level signal and a step-level signal formed by actions sharing an anchor state (Feng et al., 2026). More broadly, reinforcement learning has been applied to multi-turn LLM agents through hierarchical value aggregation, search-guided exploration, and multi-turn rollouts (Zhou et al., 2024b; Putta et al., 2024; Chen et al., 2025; Wang et al., 2025). SGPO retains this two-level construction and changes only the return placed inside the step-level normalization.

State representation for stepwise credit. An observation may alias histories that require different actions. HGPO addresses this issue through hierarchical groups defined by increasingly detailed historical contexts (He et al., 2026). That approach changes the partition used for comparison; SGPO instead holds the anchor partition fixed and changes how the future return of each row is estimated. The two mechanisms target complementary errors—state aliasing and long-suffix return variance—but we evaluate them as separate baselines here.

For bootstrapped value estimation, temporal-difference learning and TD(λ) interpolate between bootstrapping and Monte Carlo evaluation (Sutton, 1988; Sutton et al., 1998), and underlie value-function approximation in deep reinforcement learning (Mnih et al., 2015). GAE uses an analogous trace for actor–critic advantages (Schulman et al., 2015), while least-squares TD methods solve batch fixed-point systems under a chosen representation (Bradtke & Barto, 1996). SGPO uses a tabular representation induced by anchor states and solves a fresh empirical Bellman system per rollout group. No value parameters persist across batches, and the solution is used only to construct the current policy update; related work for LLM reasoning likewise refines credit assignment by replacing learned value estimates with fresh Monte Carlo rollout targets (Kazemnejad et al., 2024). Graph-based state representations have also been used for text-game agents (Ammanabrolu & Riedl, 2019), building on deep reinforcement learning for language-based control (Narasimhan et al., 2015). A closely related method also organizes group rollouts into a state-transition graph for long-horizon agents (Wang et al., 2026), forming an edge-centric advantage over a global graph; SGPO instead solves a per-group certainty-equivalent Bellman system and uses its value only as the TD(λ) bootstrap target, discarding it after each group.

3 Preliminaries

For task x, $s^{\mathrm{env}}_{h,t}$ and $o_{h,t}$ denote the underlying environment state and output; the full state need not be observed. Rollouts have a finite step cap and discount $0 \le \gamma < 1$. The complete logged history $H_{h,t}$ before $a_{h,t}$ includes the task, environment outputs, anchor metadata, past actions and rewards, and elapsed steps. The formatted, possibly truncated policy input is $c_{h,t} = \psi(H_{h,t})$; the environment-defined `anchor_obs` key is $\sigma_{h,t} = \phi(H_{h,t}) \in \Sigma$. Equal keys need not imply equal environment states or policy contexts. The behavior policy $\beta = \pi_{\theta_{\mathrm{old}}}$ samples $N \ge 2$ trajectories $\mathcal{G}_g = \{\tau_h\}_{h=1}^{N}$ from matched initial conditions, where $\tau_h = (o_{h,0}, a_{h,0}, r_{h,0}, \ldots, o_{h,T_h})$. A *row* $i = (h,t)$ is one action and environment response; $i+1$ always refers to the next row of the same trajectory. Episode returns and advantages are

$$R_h = \sum_{t=0}^{T_h-1} \gamma^t r_{h,t}, \qquad A^{\mathrm{epi}}_h = \frac{R_h - \frac{1}{N}\sum_{j=1}^{N} R_j}{\mathrm{std}_{j\in\{1,\ldots,N\}}(R_j) + \epsilon}. \tag{1}$$

With step weight $w_{\mathrm{step}} \ge 0$, the combined advantage is

$$A_{h,t} = A^{\mathrm{epi}}_h + w_{\mathrm{step}} A^{\mathrm{step}}_{h,t}. \tag{2}$$

Within a task group, rows with identical anchor strings form I_σ. For scalar targets z_i, define

$$\mathrm{Norm}_\sigma(z_i) = \frac{z_i - \mathrm{mean}_{j\in I_\sigma} z_j}{\mathrm{std}_{j\in I_\sigma} z_j + \epsilon}, \quad I_\sigma = \{j : \sigma_j = \sigma\}. \tag{3}$$

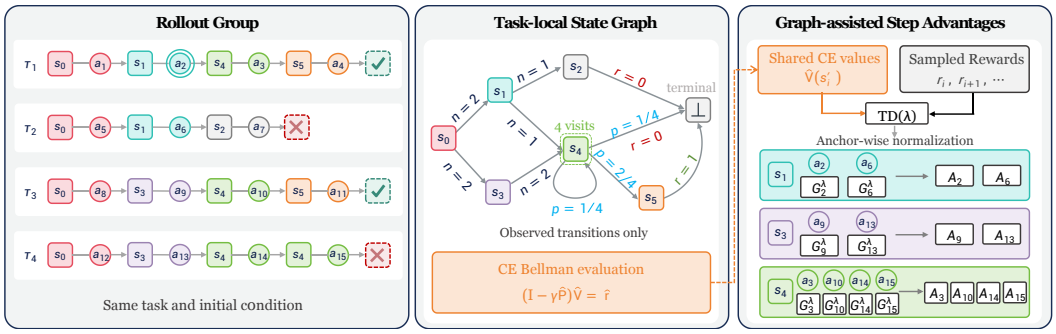


Figure 2: The SGPO pipeline. Left: rollouts from one task and initial condition. Middle: an empirical graph over anchors s_j, with counts n, successor probabilities p, rewards r, and terminal $\perp$; CE evaluation solves $(I - \gamma\widehat{P})\widehat{V} = \widehat{r}$ on active anchors. Right: shared successor values and sampled rewards yield TD(λ) returns G_j^λ, normalized within anchors into step advantages A_j for actions a_j. Here A_j denotes A_i^{step}, not the combined advantage.

Here $\epsilon > 0$ stabilizes division; std is the sample standard deviation, and singleton anchors output zero. GIGPO uses $A_i^{\text{step}} = \text{Norm}_{\sigma_i}(G_i^{\text{MC}})$, where G_i^{MC} is the discounted return along row i's own trajectory suffix.

4 State-Graph Policy Optimization

4.1 Framework Overview

Monte Carlo step returns use a single sampled continuation, although other trajectories in the same rollout group may reach shared successor anchors and provide additional reward evidence. SGPO pools this evidence in a task-local graph. Policy context c_i governs action generation and likelihoods; anchor σ_i defines graph nodes and comparison groups, as in Section 3.

Figure 2 pools observed transitions and rewards from one task and initial condition into an empirical anchor graph. Its certainty-equivalence (CE) Bellman solution supplies shared successor values; these and each trajectory's sampled rewards are the two inputs to backward TD(λ). The reward sequences and current-anchor groups remain unchanged. Subtracting the current-anchor value and normalizing within its group yields a step advantage, combined with the episode advantage via equation 2 for the policy update.

Only the return used in step normalization changes:

$$G_i^{\text{MC}} \longrightarrow G_i^{\lambda}. \tag{4}$$

Rollout sampling, policy contexts, anchor partitions, episode advantages, and outer optimization remain unchanged. Each group's graph and values are computed and discarded locally, without a persistent critic, additional rollouts, or extra model evaluations.

4.2 State-Graph Step Relative Advantage

Each row $i = (h, t)$ gives an edge $\sigma_i \xrightarrow{r_i} \sigma_i'$, with $\sigma_i' = \phi(H_{h,t+1}) = \sigma_{i+1}$ for continuing trajectories and $\sigma_i' = \perp$, $\widehat{V}(\perp) = 0$ at termination or step-cap truncation. The target is therefore the capped-episode return. Active anchors have outgoing rows; let $n(\sigma)$ and $n(\sigma, \sigma')$ count these rows and transitions:

$$\widehat{r}(\sigma) = \frac{1}{n(\sigma)} \sum_{i:\sigma_i=\sigma} r_i, \qquad \widehat{P}(\sigma' \mid \sigma) = \frac{n(\sigma, \sigma')}{n(\sigma)}. \tag{5}$$

Counts pool only one task-conditioned group, never different tasks, environment seeds, or batches. Padding duplicates are removed before counting and inherit their original row's advantage afterward.

Figures 1(c) and 2 illustrate sharing at s_4: four outgoing rows comprise two transitions to s_5, one self-loop, and one terminal transition. Hence $\widehat{P}(s_5 \mid s_4) = 2/4$ and $\widehat{P}(s_4 \mid s_4) = \widehat{P}(\perp \mid s_4) = 1/4$. These observations inform $\widehat{V}(s_4)$, which bootstraps $s_1 \xrightarrow{a_2} s_4$. The s_1 comparison remains between a_2 and a_6; actions a_9 and a_{13} from s_3 do not join it.

Pooling sampled actions and histories yields an empirical Markov reward process $(\widehat{r}, \widehat{P})$ on anchors; $\widehat{P}$ is not an action-conditioned environment kernel. Its CE Bellman system is

$$\widehat{V}(\sigma) = \widehat{r}(\sigma) + \gamma \sum_{\sigma' \neq \perp} \widehat{P}(\sigma' \mid \sigma)\widehat{V}(\sigma'), \qquad (I - \gamma\widehat{P})\widehat{V} = \widehat{r}. \tag{6}$$

Successors are averaged by frequency, not maximized. The solution is the empirical anchor-model value, not necessarily the value conditioned on c_i or H_i; a policy-value interpretation requires Proposition 2.

Graph-assisted TD(λ) return. For $0 \leq \lambda \leq 1$, recurse backward along each sampled trajectory:

$$G_i^\lambda = r_i + \gamma \left[(1-\lambda)\widehat{V}(\sigma_i') + \lambda G_{i+1}^\lambda \right], \tag{7}$$

with zero continuation after the last row. The endpoints are one-step CE ($\lambda = 0$) and Monte Carlo ($\lambda = 1$); intermediate λ geometrically mixes sampled rewards and graph bootstraps. The residual and step advantage are

$$\widetilde{A}_i = G_i^\lambda - \widehat{V}(\sigma_i), \qquad A_i^{\text{step}} = \text{Norm}_{\sigma_i}(\widetilde{A}_i). \tag{8}$$

The anchor-constant baseline cancels in equation 3; graph values affect advantages through successor bootstrapping in equation 7. The default $\lambda = 0.75$ is fixed on a development split, not the test set. One step advantage is returned per input row, in order (implementation conventions in Appendix D).

4.3 Policy Optimization and Complexity

Each ratio uses its row's $c_i = \psi(H_i)$ regardless of shared anchors. For clipping width $\varepsilon > 0$, the row-level surrogate is

$$J(\theta) = \mathbb{E}_i \left[\min(\rho_i(\theta)A_i, \text{clip}(\rho_i(\theta), 1-\varepsilon, 1+\varepsilon)A_i) \right], \quad \rho_i(\theta) = \frac{\pi_\theta(a_i \mid c_i)}{\pi_{\theta_{\text{old}}}(a_i \mid c_i)}. \tag{9}$$

Anchor-normalized advantages need not be unbiased history-conditioned policy advantages. Row weighting and KL/entropy regularizers remain unchanged. For m rows and S active anchors, counting and TD recursion each cost $O(m)$; the dense solve costs $O(S^3)$, adding no model parameters or critic activations (timing and peak GPU memory in Appendix **??**).

4.4 Estimator Properties and Error Analysis

Proposition 1 (well-posedness). The active-anchor matrix $\widehat{P}$ is substochastic. For $0 \leq \gamma < 1$, $I - \gamma\widehat{P}$ is nonsingular and

$$\widehat{V} = (I - \gamma\widehat{P})^{-1}\widehat{r} = \sum_{k=0}^{\infty} \gamma^k \widehat{P}^k \widehat{r}, \qquad \|(I - \gamma\widehat{P})^{-1}\|_\infty \leq \frac{1}{1-\gamma}. \tag{10}$$

At $\gamma = 1$, invertibility requires spectral radius $\rho(\widehat{P}) < 1$. For $\gamma < 1$ (used in the implementation), well-posedness does not require Markov anchors.

Proposition 2 (estimation target). Fix β, a finite anchor space, and bounded rewards. Assume i.i.d. capped rollouts of one task instance. Under $a \sim \beta(\cdot \mid \psi(H))$, suppose the conditional mean reward and successor-anchor distribution depend on H only through $\phi(H)$, including termination and the step cap; denote them r^β and P^β. As rollout count grows, with every anchor having positive visit probability, $\widehat{r} \to r^\beta$, $\widehat{P} \to P^\beta$, and $\widehat{V} \to V^\beta = (I - \gamma P^\beta)^{-1} r^\beta$ almost surely. This is consistency for an anchor-level behavior value under a fixed policy, not a consequence of exact key matching or shared environment dynamics.

Theorem 1 (Monte Carlo endpoint). With Monte Carlo and TD returns computed on fixed rollout rows under matched discounts, terminal/truncation conventions, anchor partitions, and normalization, $\lambda = 1$ yields $A_i^{\text{step}} = \text{Norm}_{\sigma_i}(G_i^{\text{MC}})$: equation 7 reduces to G_i^{MC}, and the baseline cancels in equation 3. This identity requires neither Markov anchors nor equal policy contexts. Appendix A proves it; the implementation also checks the endpoint numerically.

For $\delta_i = r_i + \gamma \widehat{V}(\sigma_i') - \widehat{V}(\sigma_i)$ and $T_i = T_h - t$ remaining rows in the same trajectory, unrolling gives

$$G_i^{\lambda} - \widehat{V}(\sigma_i) = \sum_{k=0}^{T_i - 1} (\gamma\lambda)^k \delta_{i+k}. \tag{11}$$

Smaller λ shortens the sampled trace but increases reliance on the estimated model; it does not guarantee lower variance. Under Proposition 2's abstraction condition, on a common finite anchor space, define $\varepsilon_r = \|\widehat{r} - r^{\beta}\|_{\infty}$ and $\varepsilon_P = \|\widehat{P} - P^{\beta}\|_{\infty}$ (maximum absolute row sum). Then

$$\|\widehat{V} - V^{\beta}\|_{\infty} \leq \frac{\varepsilon_r + \gamma \varepsilon_P \|V^{\beta}\|_{\infty}}{1 - \gamma}. \tag{12}$$

Non-Markov anchors require an additional Bellman-residual term for comparison with history-conditioned values (Appendix A.5). These bounds motivate the fixed-λ sweep and graph-value controls.

5 Experiments

We compare policy performance under matched protocols, then ablate graph bootstrapping, the trace parameter, and episode-level credit. We also test how graph connectivity affects held-out prediction and online learning, analyze signal quality across training stages and graph coverage, and measure the estimator's time and memory costs.

5.1 Experimental Setup

Benchmarks. We use two long-horizon benchmarks with sparse terminal rewards. ALFWorld (Shridhar et al., 2020) is a text-based embodied environment with six household task categories (Pick, Clean, Cool, Look, Heat, and Pick2) requiring state perception, planning, and multi-step actions. WebShop (Yao et al., 2022a) contains over one million products and roughly twelve thousand instructions; agents search, navigate product pages, and select attributes to satisfy natural-language requirements. In both environments, tens of decisions may precede end-of-episode feedback. Task configurations are in Appendix C.1.

Baselines. Closed-source and prompting references include GPT-4o (Hurst et al., 2024), Gemini-2.5-Pro (Comanici et al., 2025), ReAct (Yao et al., 2022b), and Reflexion (Shinn et al., 2023). RL baselines include critic-based PPO (Schulman et al., 2017), critic-free RLOO (Kool et al., 2019; Ahmadian et al., 2024) and GRPO (Shao et al., 2024), and step-level group methods GiGPO (Feng et al., 2026) and HGPO (He et al., 2026). GiGPO retains the two-level grouping but uses Monte Carlo suffix returns; HGPO changes the history context used to define step groups. The two baselines let us compare changes to return estimation and group partitioning.

Implementation details. Following Feng et al. (2026), we use Qwen2.5-Instruct (1.5B and 7B) (Qwen et al., 2025), with a same-protocol Qwen3.5-2B comparison in Appendix C.2. Matched training and evaluation use eight trajectories per group, $\gamma = 0.95$, learning rate 10^{-6}, fixed SGPO $\lambda = 0.75$, and evaluation temperature 0.4. We report the final checkpoint rather than selecting the peak. Shared settings, exact model revisions, and environment-specific configurations are in Appendix C.1.

5.2 Main Results

Table 1 compares Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct; the Qwen3.5-2B comparison is in Appendix C.2, Table 3.

Table 1: Test performance on ALFWorld and WebShop. For ALFWorld we report the success rate (%) for each sub-task and the overall result; for WebShop we report the task score and the success rate (%). Boldface marks the best result in each column within each block. RL-training entries are three-seed means with sample standard deviations; prompting entries are single-run references.

Type	Method	ALFWorld							WebShop	
		Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Succ.
Closed-Source Models										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	**92.8**	**63.3**	**62.1**	**69.0**	**26.6**	**58.7**	**60.3**	**42.5**	**35.9**
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training	PPO (with critic)	$64.8_{\pm 3.5}$	$57.1_{\pm 4.9}$	$46.4_{\pm 4.0}$	$40.5_{\pm 6.9}$	$60.6_{\pm 6.6}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$	$73.8_{\pm 3.0}$	$51.5_{\pm 2.9}$
RL Training	RLOO	$88.3_{\pm 3.0}$	$52.8_{\pm 8.6}$	$71.0_{\pm 5.9}$	$62.8_{\pm 8.7}$	$66.4_{\pm 5.5}$	$56.9_{\pm 4.7}$	$69.7_{\pm 2.5}$	$73.9_{\pm 5.6}$	$52.1_{\pm 6.7}$
RL Training	GRPO	$85.3_{\pm 1.5}$	$53.7_{\pm 8.0}$	$84.5_{\pm 6.8}$	$78.2_{\pm 7.9}$	$59.7_{\pm 5.0}$	$53.5_{\pm 5.6}$	$72.8_{\pm 3.6}$	$75.8_{\pm 3.5}$	$56.8_{\pm 3.8}$
RL Training	GiGPO	$\mathbf{94.4}_{\pm 5.9}$	$94.8_{\pm 3.8}$	$79.8_{\pm 4.7}$	$67.5_{\pm 4.6}$	$94.4_{\pm 7.8}$	$76.4_{\pm 5.4}$	$86.7_{\pm 1.7}$	$83.1_{\pm 1.6}$	$65.0_{\pm 3.2}$
RL Training	HGPO	$94.3_{\pm 2.9}$	$95.9_{\pm 2.8}$	$\mathbf{92.0}_{\pm 2.3}$	$69.1_{\pm 9.2}$	$\mathbf{97.4}_{\pm 3.6}$	$82.3_{\pm 3.2}$	$90.5_{\pm 0.4}$	$85.5_{\pm 0.5}$	$70.5_{\pm 1.7}$
RL Training	SGPO	$92.7_{\pm 3.9}$	$\mathbf{97.1}_{\pm 1.2}$	$90.3_{\pm 1.3}$	$\mathbf{80.9}_{\pm 3.0}$	$96.9_{\pm 2.5}$	$\mathbf{97.0}_{\pm 2.3}$	$\mathbf{93.6}_{\pm 0.6}$	$\mathbf{89.6}_{\pm 0.7}$	$\mathbf{81.5}_{\pm 1.9}$
Qwen2.5-7B-Instruct										
Prompting	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
RL Training	PPO (with critic)	$92.3_{\pm 4.0}$	$92.5_{\pm 2.4}$	$80.3_{\pm 2.0}$	$64.0_{\pm 8.4}$	$89.5_{\pm 7.0}$	$68.8_{\pm 8.3}$	$80.4_{\pm 2.7}$	$81.4_{\pm 3.1}$	$68.7_{\pm 5.1}$
RL Training	RLOO	$87.6_{\pm 4.3}$	$78.2_{\pm 8.3}$	$87.3_{\pm 5.8}$	$81.3_{\pm 7.6}$	$71.9_{\pm 5.2}$	$48.9_{\pm 8.4}$	$75.5_{\pm 4.6}$	$80.3_{\pm 3.2}$	$65.7_{\pm 4.0}$
RL Training	GRPO	$90.8_{\pm 5.1}$	$66.1_{\pm 6.7}$	$\mathbf{89.3}_{\pm 5.4}$	$74.7_{\pm 6.9}$	$72.5_{\pm 5.4}$	$64.7_{\pm 7.3}$	$77.6_{\pm 5.2}$	$79.3_{\pm 2.8}$	$66.1_{\pm 3.7}$
RL Training	GiGPO	$97.7_{\pm 1.6}$	$\mathbf{98.8}_{\pm 1.6}$	$\mathbf{89.3}_{\pm 8.2}$	$82.7_{\pm 7.9}$	$83.7_{\pm 7.2}$	$79.2_{\pm 6.6}$	$90.8_{\pm 1.3}$	$84.4_{\pm 2.9}$	$72.8_{\pm 3.2}$
RL Training	HGPO	$\mathbf{99.7}_{\pm 0.5}$	$95.0_{\pm 1.4}$	$82.1_{\pm 9.1}$	$84.0_{\pm 1.8}$	$94.0_{\pm 2.4}$	$86.1_{\pm 5.3}$	$91.8_{\pm 1.8}$	$86.0_{\pm 0.7}$	$72.8_{\pm 1.6}$
RL Training	SGPO	$98.3_{\pm 1.1}$	$96.0_{\pm 1.3}$	$83.6_{\pm 4.1}$	$\mathbf{93.6}_{\pm 2.0}$	$\mathbf{96.7}_{\pm 3.9}$	$\mathbf{93.2}_{\pm 1.1}$	$\mathbf{94.7}_{\pm 0.4}$	$\mathbf{92.1}_{\pm 0.5}$	$\mathbf{83.7}_{\pm 0.8}$

Main-result analysis. At 1.5B/7B, SGPO achieves the highest ALFWorld overall success (93.6/94.7) and WebShop score/success (89.6/81.5 and 92.1/83.7). ALFWorld gains over the best baseline, HGPO (90.5/91.8), are 3.1 and 2.9 percentage points. WebShop success gains are larger: 11.0 over HGPO and 16.5 over GiGPO at 1.5B, and 10.9 over both GiGPO and HGPO at 7B. Gains on ALFWorld tasks with longer suffixes and more compositional structure offset losses on some shorter tasks. These gains and the larger WebShop margins are consistent with better credit assignment over long suffixes; the comparison groups are unchanged. Results on Qwen3.5-2B follow the same trend (Appendix C.2).

5.3 Ablations and Graph-Value Controls

Ablations, offline analyses, and learning dynamics use Qwen2.5-1.5B-Instruct, with matched initial weights, prompts, rewards, state keys, group sampling, and optimization. Metrics remain ALFWorld sub-task/overall success and WebShop score/success.

λ sensitivity. We sweep $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$: 0 is one-step CE, 1 is the matched GiGPO Monte Carlo endpoint, and 0.75 is the default SGPO setting. Intermediate settings outperform both endpoints (Figure 3). On WebShop, $\lambda = 0$ and 1 yield score/success 85.8/78.1 and 83.1/65.0, below the default's 89.6/81.5. WebShop success is particularly sensitive to the trace, rising from 65.0 to 81.5. ALFWorld overall success is also higher at intermediate values: 94.5 at $\lambda = 0.25$ and 93.6 at 0.75, versus 89.8 at 0 and 86.7 at 1. At $\lambda = 1$, the per-task success rates are Pick 94.4, Clean 94.8, Cool 79.8, Look 67.5, Heat 94.4, and Pick2 76.4, whereas graph bootstrapping at 0.75 improves every listed sub-task. The largest absolute gains are on Look (+13.4) and Cool (+10.5), while Clean and Heat also improve modestly. These task-level gains suggest improved credit assignment on longer, more compositional tasks. Intermediate settings perform well, but we do not interpret small differences between nearby values as statistically significant. We fixed the default on development data and did not reselect it on the test set.

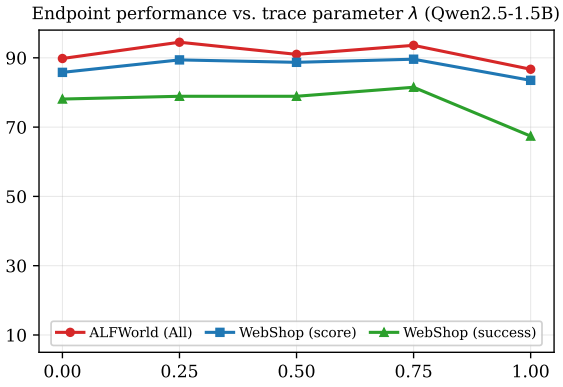


Figure 3: ALFWorld overall success and WebShop score/success (%) versus the trace parameter λ on Qwen2.5-1.5B-Instruct.

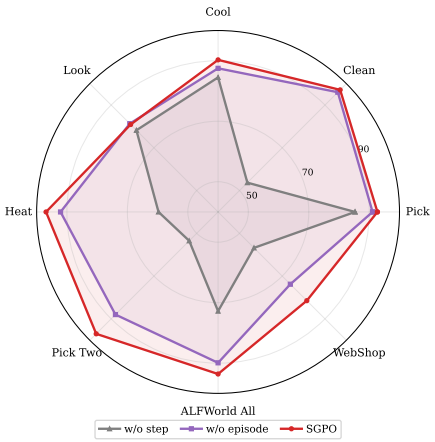


Figure 4: Advantage-component ablation on SGPO.

Advantage components. In both environments, we compare full SGPO with w/o episode ($\lambda = 0.75$, $A_i = w_{\text{step}} A_i^{\text{step}}$) and w/o step ($A_i = A_{h(i)}^{\text{epi}}$, equivalent to matched GRPO). Removing either term degrades performance (Figure 4). Without episode credit, ALFWorld overall success falls to 89.9 from 93.6, and WebShop score/success to 85.7/73.8 from 89.6/81.5. Step credit alone therefore retains most of the ALFWorld performance; adding episode credit helps more on WebShop. Removing step credit causes larger drops in both environments (ALFWorld 72.8; WebShop 75.8/56.8). The transition-aware step signal accounts for most of the gains in this ablation.

Online transition intervention. Does correct connectivity matter beyond using a bootstrap? On WebShop, Shuffled CE ($\lambda = 0.75$, three seeds) permutes nonterminal successor labels within each rollout group, preserving sources, rewards, terminal/truncation edges, sample counts, and the successor-label multiset. CE is fit on the rewired graph, while TD returns follow real trajectories and query real successors. Score/success falls to 82.9/70.3 from full SGPO's 89.6/81.5, a drop of 6.7 score points and 11.2 success percentage points. The shuffled control is closer to the Monte Carlo/GIGPO endpoint (83.1/65.0), though its success remains higher. Retaining bootstrapping but disrupting connectivity reduces performance, supporting a benefit from informative transitions beyond bootstrapping alone (Appendix C.3).

Held-out prediction and coverage. Leakage-free leave-one-trajectory-out prediction provides complementary evidence: Real CE has the lowest pooled MSE among MC, Local, Zero, and Shuffled CE in both environments under both behavior sources. Its advantage over Local and Zero supports value propagation beyond suffix shortening, but is not a direct measurement of policy-gradient variance. With the logged policy and held-out trajectory fixed, nested fitting sets of 3/5/7 trajectories separately probe limited graph coverage. Coverage sets differ across sizes, so raw MSE changes do not isolate a sample-size effect (Appendix C.4).

5.4 Learning Dynamics and Step-Signal Diagnostics

Figure 5 compares one-seed Qwen2.5-1.5B-Instruct rollout success for default $\lambda = 0.75$ and Monte Carlo $\lambda = 1$. The default reaches higher success earlier in both environments. These runs are separate from the three-seed main experiments.

Step-signal diagnostics. The default has fewer groups with all-zero step advantages early in training. On WebShop at iteration 10, this fraction is 50.0%, versus 87.5% for $\lambda = 1$; the nonzero step-advantage fraction is 36.4% versus 9.1% (Table 6). These observations describe the availability of step-level feedback, not its correctness: an all-zero step term does not imply zero episode-level credit, and more nonzero advantages do not establish better estimates. The policies evolve independently, and endpoints and AUC windows are not fully matched, so the curves and diagnostics are descriptive,

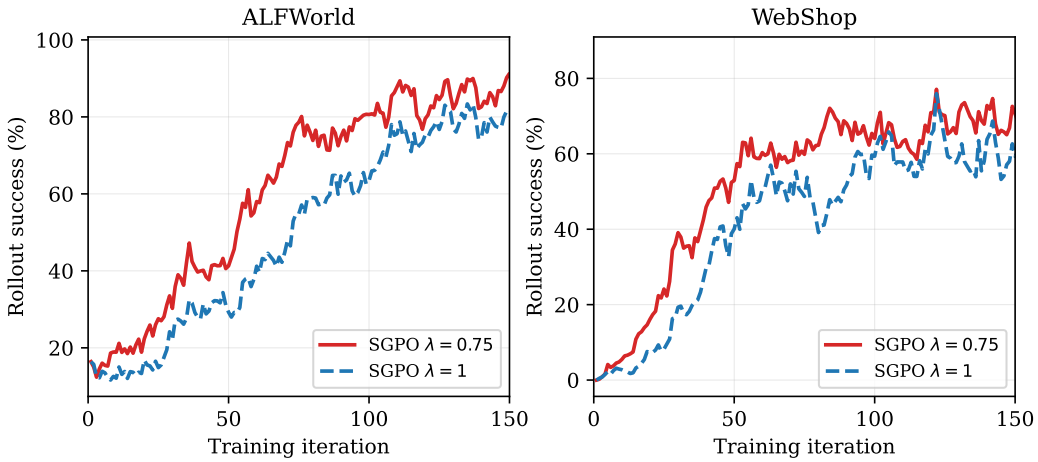


Figure 5: One-seed rollout success over training on Qwen2.5-1.5B-Instruct: the default graph-assisted return ($\lambda = 0.75$) versus the Monte Carlo step-return ($\lambda = 1$). Left: ALFWorld; right: WebShop. Last-logged values and AUC summaries are in Table 7; endpoints and integration windows across streams are not fully matched.

not substitutes for the three-seed main comparison. Full graph diagnostics, AUC summaries, and illustrative return-estimation cases are in Appendix C.5. Exploratory visual-backbone and online group-size checks are reported separately in Appendix C.6.

6 Limitations

SGPO pools trajectories sharing task-conditioned dynamics; cross-task pooling changes the target. Exact-string anchors trade coverage against aliasing; non-Markov anchors require an additional abstraction-error term (Appendix A.5). Same-batch value fitting and state reweighting through local standardization prevent directly invoking unbiased policy-gradient guarantees. The development-selected $\lambda = 0.75$ need not be optimal elsewhere. Main comparisons, trace sweeps, and connectivity controls use three training seeds; offline evidence uses one logged stream per environment/source, while single-seed visual-transfer and smaller-group checks remain directional.

7 Conclusion

SGPO combines certainty-equivalent evaluation of task-local empirical graphs with real-trajectory TD(λ), keeping anchor groups and the outer optimizer fixed without a learned critic, extra rollouts, or model evaluations. Under matched conventions, $\lambda = 1$ exactly recovers GiGPO's Monte Carlo step advantage. Across two model scales on ALFWorld and WebShop, SGPO improves mean aggregate performance over the strongest evaluated group-based baselines. Online shuffling supports informative connectivity beyond bootstrapping alone; leakage-free offline controls support value propagation beyond suffix shortening. Prediction gains concentrate early/mid-training, with late reversals on WebShop, and do not establish lower policy-gradient variance. Exploratory Sokoban results suggest visual applicability, not a generalization guarantee. Future work includes less aliased anchors, uncertainty-aware traces, and pooling beyond one rollout group.

Ethics Statement

The method adds neither data collection nor new model capabilities, but more efficient agent training may amplify downstream benefits or harms. Our experiments use simulated household and shopping environments and avoid personal data; deployment would additionally require action constraints, monitoring, human authorization, and attention to environment licenses and compute consumption.

REPRODUCIBILITY STATEMENT

Appendices C and D specify configurations, metrics, implementation conventions, and validation checks. We will release anonymous code, resolved configurations, per-seed logs, checkpoints, task manifests, and held-out folds.

AI USE STATEMENT

We used generative AI to draft and revise prose and format tables and references. The authors conceived and carried out the methodology, theory, proofs, and experimental design; AI assistance on these parts was limited to editing and exposition. We use ALFWorld and WebShop without synthetic-dataset generation. The authors reviewed all AI-assisted work and take responsibility for the text, claims, proofs, and experimental artifacts of the paper.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Prithviraj Ammanabrolu and Mark Riedl. Playing text-adventure games with graph-based deep reinforcement learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3557–3565, 2019.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Shuo He, Lang Feng, Xin Cheng, Lei Feng, Bo An, et al. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations*, volume 2026, pp. 27572–27593, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

Wouter Kool, Herke Van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1–11, 2015.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Qwen, :, 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, Tianyi Tang, 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, 2025. URL `https://arxiv.org/abs/2412.15115`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Max-Philipp B. Schrader. Gym-Sokoban. `https://github.com/mpSchrader/gym-sokoban`, 2018.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yunan Wang, Minghui Song, Zihan Zhang, Shaohan Huang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Group-graph policy optimization for long-horizon agentic reinforcement learning. *arXiv preprint arXiv:2606.22995*, 2026.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024a.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024b.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A PROOFS

A.1 PROOF OF PROPOSITION 1

The rows of the active-anchor matrix $\widehat{P}$ sum to at most one because probability mass may transition to the omitted absorbing state. Hence $\|\widehat{P}\|_\infty \leq 1$ and the spectral radius $\rho(\gamma\widehat{P}) \leq \gamma < 1$. The Neumann series $\sum_{k=0}^{\infty}(\gamma\widehat{P})^k$ converges and is the inverse of $I - \gamma\widehat{P}$. Multiplying by $\widehat{r}$ gives equation 10, and submultiplicativity yields the stated norm bound. □

A.2 PROOF OF PROPOSITION 2

The abstraction condition in Proposition 2 is, for every reachable nonterminal history H,

$$\mathbb{E}_\beta[r \mid H] = r^\beta(\phi(H)), \qquad \Pr_\beta(\sigma^+ = \sigma' \mid H) = P^\beta(\sigma' \mid \phi(H)),$$

where actions are sampled from $\beta(\cdot \mid \psi(H))$ and $\sigma^+ = \perp$ after termination or truncation. Here the full successor kernel includes $\perp$; the matrix in the Bellman system omits its row and column. Thus histories sharing an anchor must induce the same mean reward and successor-anchor law, even if their policy contexts differ. Environment-level Markov dynamics alone do not imply this condition for a history-dependent policy. The step cap can also violate it when an anchor aliases different remaining times.

Under these assumptions, each i.i.d., bounded-length rollout contributes finite counts and bounded total reward to each anchor. The strong law applied to these trajectory-level totals, followed by division by visit counts, gives $\widehat{r} \to r^\beta$ and $\widehat{P} \to P^\beta$ almost surely for every visited anchor. On the fixed finite anchor space, inversion is continuous at $I - \gamma P^\beta$, so $\widehat{V} \to (I - \gamma P^\beta)^{-1} r^\beta = V^\beta$. Let $U^\beta(H)$ denote the expected discounted remaining capped return under the context-conditioned policy. The abstraction condition makes $V^\beta(\phi(H))$ satisfy the history Bellman equation, so uniqueness gives $U^\beta(H) = V^\beta(\phi(H))$. Without that condition, the practical key need not preserve the history value. □

A.3 PROOF OF THEOREM 1

At $\lambda = 1$, equation 7 becomes $G_i^1 = r_i + \gamma G_{i+1}^1$. Backward induction from the terminal row yields the ordinary discounted Monte Carlo return G_i^{MC}. For all $i \in I_\sigma$, $\widehat{V}(\sigma_i) = \widehat{V}(\sigma)$ is constant. Subtracting a constant does not change the centered values or standard deviation of a sample, so $\mathrm{Norm}_\sigma(G_i^1 - \widehat{V}(\sigma)) = \mathrm{Norm}_\sigma(G_i^{\mathrm{MC}})$. Thus the step advantage equals the anchor-normalized Monte Carlo return. □

A.4 DERIVATION OF THE TD(λ) TRACE

Subtract $\widehat{V}(\sigma_i)$ from equation 7 and introduce $\delta_i = r_i + \gamma\widehat{V}(\sigma_i') - \widehat{V}(\sigma_i)$. For consecutive sampled rows, $\sigma_i' = \sigma_{i+1}$, so

$$\begin{aligned} G_i^\lambda - \widehat{V}(\sigma_i) &= \delta_i + \gamma\lambda\left(G_{i+1}^\lambda - \widehat{V}(\sigma_{i+1})\right) \\ &= \sum_{k=0}^{T_i-1}(\gamma\lambda)^k\delta_{i+k}, \end{aligned}$$

Here T_i counts remaining rows within the same trajectory. Both continuation return and terminal value are zero after its last row, yielding equation 11.

A.5 VALUE ERROR AND NON-MARKOV ANCHORS

On the common Markov anchor space, subtracting the two Bellman systems gives

$$\widehat{V} - V^\beta = (I - \gamma\widehat{P})^{-1}\left[(\widehat{r} - r^\beta) + \gamma(\widehat{P} - P^\beta)V^\beta\right].$$

Proposition 1 and the induced matrix infinity norm yield equation 12; empirical rewards must be defined on this common space, so the bound does not by itself resolve missing-anchor coverage.

For the following uniform history-value bound, assume both models are defined on a common finite anchor space containing every anchor reachable under β; the bound does not apply as written to uncovered anchors. For non-Markov anchors, fix an anchor surrogate $(\bar{r}, \bar{P})$ with substochastic $\bar{P}$ and value $\bar{V} = (I - \gamma\bar{P})^{-1}\bar{r}$ and lift it to histories as $v(H) = \bar{V}(\phi(H))$, extending $\phi(H) = \perp$ and $v(H) = 0$ at termination or the step cap. Let $U^\beta(H)$ be the history-conditioned capped-episode value and $(\mathcal{T}^\beta u)(H) = \mathbb{E}_\beta[r + \gamma u(H') \mid H]$ its Bellman operator, with actions sampled from $\beta(\cdot \mid \psi(H))$. Define the abstraction residual $\eta_\phi = \|\mathcal{T}^\beta v - v\|_\infty$, over reachable histories, including time-to-cap effects. Contraction gives $\|v - U^\beta\|_\infty \le \eta_\phi/(1-\gamma)$. Hence, with $\bar{\varepsilon}_r = \|\widehat{r} - \bar{r}\|_\infty$ and $\bar{\varepsilon}_P = \|\widehat{P} - \bar{P}\|_\infty$ on that common anchor space,

$$\|\widehat{V} \circ \phi - U^\beta\|_\infty \le \frac{\bar{\varepsilon}_r + \gamma\bar{\varepsilon}_P\|\bar{V}\|_\infty + \eta_\phi}{1-\gamma}.$$

This separates empirical-model error from abstraction error; exact string matching does not bound η_ϕ, and neither bound establishes lower policy-gradient variance.

B Estimator Pseudocode

Algorithm 1 details the step-return estimator for one task-conditioned rollout group. Graph values are computed once per group; the return recursion follows each sampled trajectory independently.

Algorithm 1 SGPO step-return estimator

Require: Sampled rows $(\sigma_i, \sigma_i', r_i, h_i, t_i)$, discount $0 \le \gamma < 1$,
trace weight $0 \le \lambda \le 1$, and padding identifiers
// Row preparation
Remove padding duplicates; retain a map to their original rows
Redirect terminal and truncated successors to $\perp$; set $\widehat{V}(\perp) \leftarrow 0$
// Graph aggregation and certainty-equivalent evaluation
Group rows by exact anchor key to form I_σ
Compute $\widehat{r}$ and $\widehat{P}$ from reward sums and transition counts using equation 5
Solve $(I - \gamma\widehat{P})\widehat{V} = \widehat{r}$ on active anchors
// Trajectory-wise return computation
for each trajectory h **do**
  $G^\lambda_{h,T_h} \leftarrow 0$
  for $t = T_h - 1, \dots, 0$ **do**
    $i \leftarrow (h, t)$
    $G^\lambda_{h,t} \leftarrow r_i + \gamma\big[(1-\lambda)\widehat{V}(\sigma_i') + \lambda G^\lambda_{h,t+1}\big]$
    $\widetilde{A}_i \leftarrow G^\lambda_{h,t} - \widehat{V}(\sigma_i)$
  end for
end for
// Anchor-relative advantage computation
for each anchor group I_σ **do**
  $A_i^{\text{step}} \leftarrow \text{Norm}_\sigma(\widetilde{A}_i)$ for all $i \in I_\sigma$, using equation 3
end for
Copy original advantages to padding duplicates and restore input order
return one A_i^{step} for every input row

C Extended Results and Experimental Details

This appendix provides the full experimental configuration, the extended main-result comparison on the Qwen3.5-2B backbone, the offline graph-value analysis and the learning- and graph-dynamics study. Every supplementary experiment outside the main comparison uses Qwen2.5-1.5B-Instruct, except the Sokoban transfer, which uses its own visual backbone and is labeled as such.

C.1 SHARED CONFIGURATION

All model identifiers use the `Qwen/` namespace: the main comparison uses `Qwen2.5-1.5B-Instruct` and `Qwen2.5-7B-Instruct`; the extended comparison uses `Qwen3.5-2B`; Sokoban uses `Qwen2.5-VL-3B-Instruct`. Matched methods share weights, tokenizer, chat template, and generation settings.

Table 2: Shared configuration for ALFWorld and WebShop. Supplementary deviations are stated with their results.

Setting	ALFWorld	WebShop
Maximum environment steps	50	30
Policy history length	2	2
Metrics	Per-sub-task / overall success	Task score / success
Evaluation tasks / temperature	512 / 0.4	512 / 0.4
Rollouts	8 trajectories/group; 16 groups/iteration	
Discount / learning rate	$\gamma = 0.95$ / 10^{-6}	
Trace weight	$\lambda = 0.75$; sweep: $0, 0.25, 0.5, 0.75, 1$	
Episode return	Discounting as in Eq. equation 1	

C.2 ADDITIONAL MAIN-RESULT VIEWS

Table 3 extends the matched-protocol comparison to the newer Qwen3.5-2B backbone with the columns and metrics of Table 1. The RL-training rows report mean $\pm$ standard deviation.

Table 3: In-house results on the Qwen3.5-2B backbone. Columns and metrics match Table 1: ALFWorld reports per-sub-task and overall success (%); WebShop reports the task score and success (%). RL-training rows report mean $\pm$ standard deviation; prompting rows serve as references. Boldface marks the best RL-training result.

Type	Method	ALFWorld							WebShop	
		Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Succ.
Qwen3.5-2B										
Prompting	Qwen3.5	36.5	0.0	4.5	12.2	6.9	24.5	16.8	31.1	7.2
Prompting	ReAct	65.8	33.1	27.4	33.9	24.7	1.2	34.7	53.1	21.7
Prompting	Reflexion	67.4	23.7	27.3	34.3	23.2	14.2	32.5	32.5	13.0
RL Training	PPO	$77.4_{\pm 3.2}$	$78.2_{\pm 3.7}$	$81.6_{\pm 3.1}$	$65.8_{\pm 5.4}$	$90.3_{\pm 3.8}$	$73.7_{\pm 4.9}$	$78.4_{\pm 1.6}$	$78.6_{\pm 2.7}$	$66.5_{\pm 3.5}$
RL Training	RLOO	$85.6_{\pm 2.9}$	$70.7_{\pm 5.0}$	$85.1_{\pm 3.8}$	$73.4_{\pm 5.9}$	$90.2_{\pm 3.5}$	$76.8_{\pm 4.0}$	$80.1_{\pm 1.8}$	$78.1_{\pm 2.7}$	$67.4_{\pm 3.4}$
RL Training	GRPO	$80.5_{\pm 3.2}$	$75.9_{\pm 4.6}$	$90.5_{\pm 2.8}$	$77.9_{\pm 5.1}$	$91.6_{\pm 3.3}$	$82.0_{\pm 4.2}$	$82.7_{\pm 1.7}$	$79.4_{\pm 2.3}$	$70.3_{\pm 3.1}$
RL Training	GiGPO	$\mathbf{92.6}_{\pm 2.1}$	$90.9_{\pm 3.2}$	$\mathbf{90.9}_{\pm 2.6}$	$83.3_{\pm 4.3}$	$98.7_{\pm 1.2}$	$88.0_{\pm 3.5}$	$91.2_{\pm 1.2}$	$88.6_{\pm 1.4}$	$80.5_{\pm 2.2}$
RL Training	HGPO	$87.8_{\pm 2.7}$	$82.8_{\pm 3.8}$	$\mathbf{90.9}_{\pm 2.4}$	$81.3_{\pm 4.1}$	$90.0_{\pm 3.1}$	$92.0_{\pm 2.8}$	$87.2_{\pm 1.4}$	$83.1_{\pm 1.3}$	$71.1_{\pm 2.4}$
RL Training	SGPO	$91.7_{\pm 2.3}$	$\mathbf{91.9}_{\pm 2.7}$	$90.4_{\pm 2.9}$	$\mathbf{87.0}_{\pm 3.6}$	$\mathbf{99.1}_{\pm 0.8}$	$\mathbf{96.0}_{\pm 1.8}$	$\mathbf{94.5}_{\pm 1.0}$	$\mathbf{90.5}_{\pm 0.9}$	$\mathbf{85.2}_{\pm 1.6}$

C.3 ONLINE ABLATIONS AND GRAPH-VALUE CONTROLS

Section 5.3 presents the online results, including trace sensitivity (Figure 3), advantage components (Figure 4), and the shuffled-connectivity intervention. Here we retain only additional protocol details.

The online shuffled-CE control keeps every real reward and the successor-label multiset of each group but destroys the correct connectivity: within each group we independently permute the multiset of nonterminal successor labels, leaving all terminal and truncated edges fixed, and re-solve the certainty-equivalent system. The rollout and optimizer randomness are untouched, so the only change is which transitions the graph treats as having occurred. Groups with no exchangeable nonterminal edge are left unchanged, and no transition is fabricated. Section 5.3 reports the online WebShop results; Table 4 gives the complementary offline prediction errors.

C.4 LEAKAGE-FREE OFFLINE GRAPH-VALUE COMPARISON

For sampling and folds, we use two Qwen2.5-1.5B-Instruct behavior sources in each environment: the matched SGPO policy ($\lambda = 0.75$) and its MC endpoint ($\lambda = 1$, the GIGPO control). Each logged source dumps all 16 task groups at training steps $30, 60, 90, 120, 150$ (four steps for the ALFWorld $\lambda = 1$ stream, whose step-150 dump is unavailable), giving 304 groups and $304 \times 8 = 2432$ held-out folds (Table 4). A stage denotes the policy version at that dump, with the pre/post-update convention recorded. For each eight-trajectory group we hold out one trajectory in turn and fit only on the other seven, never pooling the graph across task instances, sources, or stages; the two sources are an analysis stratum and all estimators share fold IDs.

To define the prediction and target, let $\mathcal{D}_{-h}$ denote the seven fitting trajectories. Each control c (MC, Real, Local, Zero, or Shuffled) predicts using only $\mathcal{D}_{-h}$: MC uses $\lambda_c = 1$ and the other four use $\lambda_c = 0.75$, and where a value is needed $\widehat{V}^{c,-h}$ is estimated only from these fitting trajectories. The state-conditional predictor is

$$\widehat{y}^{c,-h}(\sigma) = \frac{1}{|I_\sigma^{-h}|} \sum_{i \in I_\sigma^{-h}} G_i^{\lambda_c, c, -h}, \qquad I_\sigma^{-h} = \{i \in \mathcal{D}_{-h} : \sigma_i = \sigma\}. \tag{13}$$

For held-out row (h, t) the target is its own discounted observed suffix $y_{h,t} = \sum_{j=t}^{T_h - 1} \gamma^{j-t} r_{h,j}$. No held-out reward, edge, or suffix contributes to $\widehat{y}^{c,-h}$; the held-out anchor is only a query key. This evaluates prediction at an anchor under the logged behavior distribution, not action-conditional return or on-policy value. The MC row is the anchor-averaged MC over the fitting trajectories, so its error is not identically zero by construction.

Among the exact controls, Real solves Eq. equation 6; Local uses only $\widehat{r}(\sigma)$; Zero sets all values to zero. All three set $V(\perp) = 0$. At $\lambda = 0.75$, Zero still carries the $\gamma\lambda$-weighted real reward suffix, so it equals neither the $\lambda = 1$ MC endpoint nor a reward-only estimator. For Shuffled we independently permute the multiset of nonterminal successor labels inside each fitting group, keeping all terminal and truncated edges fixed. This preserves source counts, reward observations, the successor-label multiset, and the sample count, but need not preserve the number of distinct edges or the distinct out-degree of each node. We refit $\widehat{P}$ and solve CE after each permutation, while all TD recursions still use the real trajectory order and real successors.

For coverage and metrics, eligible rows $\mathcal{E}$ are all real, non-padding held-out action rows, including terminal actions. The common covered set $\mathcal{C}$ contains rows whose current anchor occurs as a source in the seven fitting trajectories. A non-finite solve is a recorded failure, not a reason to drop the row for only one control. On the same $\mathcal{C}$ we compute

$$\mathrm{MSE}_c = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} (\widehat{y}_i^c - y_i)^2, \qquad \rho_c = \mathrm{corr}(\mathrm{rank}\, \widehat{y}^c, \mathrm{rank}\, y), \qquad \mathrm{Coverage} = 100 \frac{|\mathcal{C}|}{|\mathcal{E}|}. \tag{14}$$

Returns retain their logged reward scale, so MSE is in squared-return units. Ties use average ranks, and a constant prediction or target reports Spearman as N/A with its count rather than zero. We average pooled MSE across sources, seeds, and captured steps with equal weights. Spearman is shown per stratum, with the pooled value only a descriptive mean reported with its effective stratum count. Coverage is reported with its covered and eligible counts. For Shuffled we fix 20 independent permutations per fold, first computing the prediction error of each permutation and then averaging the metrics, without first averaging predictions into an extra ensemble. The 20 shuffles, eight overlapping folds, and individual rows are not independent training samples, so the pooled comparisons are descriptive rather than a formal test of policy-gradient variance.

To sub-sample coverage on the same log, for each held-out fold we permute the other seven trajectories once with a fixed RNG and nest the first 3/5/7 as the fitting set, for total group sizes $n = 4/6/8$. Each subset independently rebuilds the Real CE and MC predictors, with the held-out trajectory fixed, and the comparison is restricted to the rows covered at that size; the coverage of each size is listed with it, and raw MSE differences across different coverage sets are not attributed directly to sample size; $n = 8$ is the formal offline result. Error falls and coverage rises monotonically with more fitting trajectories in every environment, source, and step, and Real CE stays at or below MC at every size with its widest margin at $n = 4$. Two representative cells: ALFWorld $\lambda = 0.75$ step 90 moves (MC/Real CE) $6.94/6.44$ at $n = 4$ to $5.15/4.71$ at $n = 8$ with coverage $0.779 \to 0.881$,

and WebShop $\lambda = 0.75$ step 30 moves $14.76/13.87$ to $12.47/11.85$ with coverage $0.794 \to 0.830$. The complete size $\times$ step grid for all four streams is not tabulated here. This diagnoses the estimator under thinner coverage at a fixed policy; the online group-size sweep in Appendix C.6 is its separate policy-level counterpart.

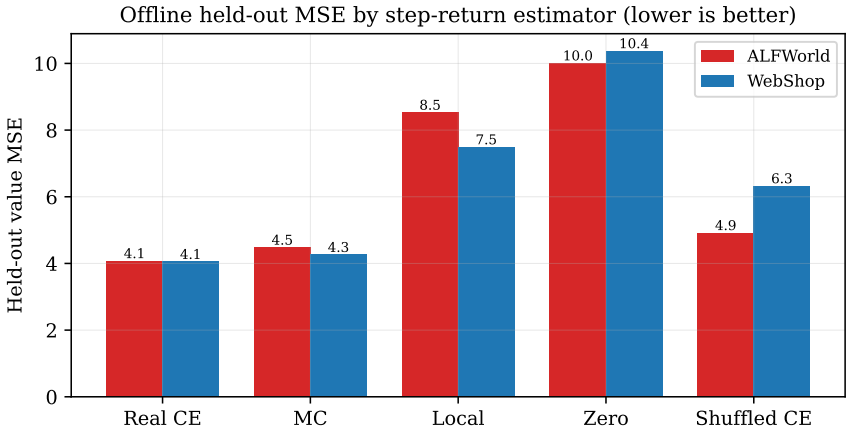


Figure 6: Held-out return MSE on the $\lambda = 0.75$ behavior stream, pooled over five captured training steps (lower is better). All estimators use the same covered held-out rows. Table 4 gives both behavior streams, Spearman ρ, coverage, and sample counts. A reduction in pooled MSE alone does not establish action-level causal credit or a reduction in policy-gradient variance.

Real CE attains the lowest MSE in both environments, and the two collapsed values separate it from a suffix-shortening artifact: Local and Zero roughly double to triple the error (WebShop 7.49 and 10.37 against 4.05), so the gain comes from propagating a graph value rather than from truncating suffixes. Shuffled CE keeps the sample counts and rewards but permutes successor labels, and falls between Real CE and the collapsed values (WebShop 6.32 against 4.05); *which* transition an edge points to therefore carries the signal. The advantage over MC is the smallest of the four pooled contrasts (ALFWorld $+0.41$, WebShop $+0.21$). Spearman ρ is nearly flat and even slightly favors the collapsed values on WebShop, so the estimators separate on calibrated magnitude, not on anchor ranking; we interpret the results as evidence for improved prediction magnitude in the pooled comparison, not uniformly better ranking or a direct measurement of policy-gradient variance.

Table 4: Offline sample counts and pooled return prediction on the four behavior streams. Each estimator cell is MSE / Spearman ρ on the same covered held-out rows; bold marks the lowest MSE in each stream. Each captured step contains 16 groups and 128 leave-one-out folds (304 groups and 2432 folds in total). Pooled MSE is reported over the captured steps; detailed per-step contrasts are omitted.

	ALFWorld		WebShop	
Behavior stream	$\lambda = 0.75$	$\lambda = 1$	$\lambda = 0.75$	$\lambda = 1$
Captured steps	30–150	30–120	30–150	30–150
Groups / folds	80 / 640	64 / 512	80 / 640	80 / 640
Coverage (%)	79.4	79.9	87.3	89.3
MC ($\lambda = 1$)	4.480 / .662	4.060 / .665	4.261 / .833	4.475 / .805
Real CE	**4.075** / .666	**3.713** / .680	**4.051** / .833	**4.357** / .796
Local	8.528 / .638	6.195 / .637	7.491 / .844	7.787 / .815
Zero	10.008 / .639	6.965 / .638	10.373 / .844	10.524 / .817
Shuffled CE	4.917 / .666	3.896 / .658	6.323 / .827	6.227 / .802

C.5 Graph and Signal Diagnostics and Learning Dynamics

This appendix reports graph and step-signal diagnostics and learning-dynamics summaries for the two traced λ settings. These streams are separate from the main comparison.

Using fixed definitions for the graph and signal diagnostics, we compute the metrics below on full eight-trajectory training groups after removing padding. Row fractions accumulate numerators and denominators before aggregation by source and stage. Each setting uses its own evolving training stream; the comparison is not a replay on identical batches.

Table 5: Fixed definitions for the graph and signal diagnostics. All fractions use real, non-padding action rows as the denominator unless stated otherwise.

Metric	Definition / use
Active node count / average distinct-successor out-degree	Excludes the absorbing node; out-degree counts observed distinct successors, not repeated edges as new
Singleton-row fraction	Real action rows whose anchor appears once, over all real action rows
Cross-trajectory successor-sharing fraction	Nonterminal rows whose successor appears as a source in another trajectory, over all real action rows; terminal rows are not in the numerator
Nonzero step-advantage fraction	Rows with $\lvert A_i^{\mathrm{step}}\rvert > 10^{-8}$ over all real action rows; threshold fixed and dtype logged
All-zero-signal group fraction	Groups where every row has $\lvert A_i^{\mathrm{step}}\rvert \le 10^{-8}$, over all groups
Step-advantage variance / episode length	Variance on the same real row set; the episode step distribution is reported separately to avoid mistaking shorter trajectories for a faster estimator

The per-iteration graph and signal dynamics are shown in Figure 7: active nodes per group, mean distinct-successor out-degree, the fraction of all-zero-signal groups, and the fraction of nonzero step advantages, for both environments under $\lambda = 0.75$ and $\lambda = 1$; Table 6 records the same quantities at iterations 10/80/160.

Table 6: Graph and signal diagnostics at iterations 10/80/160 on the Qwen2.5-1.5B-Instruct streams, using the definitions in Table 5. Each setting generates its own rollout groups. All-zero and nonzero-advantage fractions are percentages; out-degree and cross-trajectory successor-sharing fraction are dimensionless.

Environment	λ	Iter	Active nodes	Out-deg.	Succ.-sharing frac.	All-zero (%)	Nonzero adv. (%)
WebShop	0.75	10	19.7	2.06	0.25	50.0	36.4
		80	11.4	1.87	0.25	25.0	46.0
		160	11.3	1.88	0.23	6.3	61.9
	1	10	17.1	2.07	0.23	87.5	9.1
		80	13.8	2.17	0.29	37.5	33.5
		160	8.1	1.76	0.54	18.8	46.0
ALFWorld	0.75	10	56.9	2.55	0.39	43.8	42.2
		80	37.1	2.19	0.38	12.5	64.5
		160	25.3	1.96	0.50	0.0	77.5
	1	10	42.3	2.85	0.44	62.5	20.9
		80	39.4	2.41	0.39	25.0	37.8
		160	27.8	2.13	0.47	6.3	63.7

The graph and signal diagnostics reveal that the two settings differ most in the early iterations, when the reward signal is sparse. At iteration 10 the matched MC endpoint ($\lambda = 1$) leaves a large fraction of groups with a degenerate all-zero advantage: 87.5% on WebShop and 62.5% on ALFWorld, against 50.0% and 43.8% for the default $\lambda = 0.75$. The share of nonzero step advantages moves the same way, rising from 9.1% to 36.4% on WebShop and from 20.9% to 42.2% on ALFWorld once the graph bootstrap is active. The gap narrows but does not close as training proceeds: at iteration 160 the nonzero step-advantage fraction is 61.9% versus 46.0% on WebShop and 77.5% versus 63.7% on ALFWorld. Active nodes per group and average out-degree contract in both settings as the policy concentrates on shorter, more reliable trajectories, and the cross-trajectory successor-sharing fraction stays between 0.23 and 0.54, confirming that the graph repeatedly revisits the same

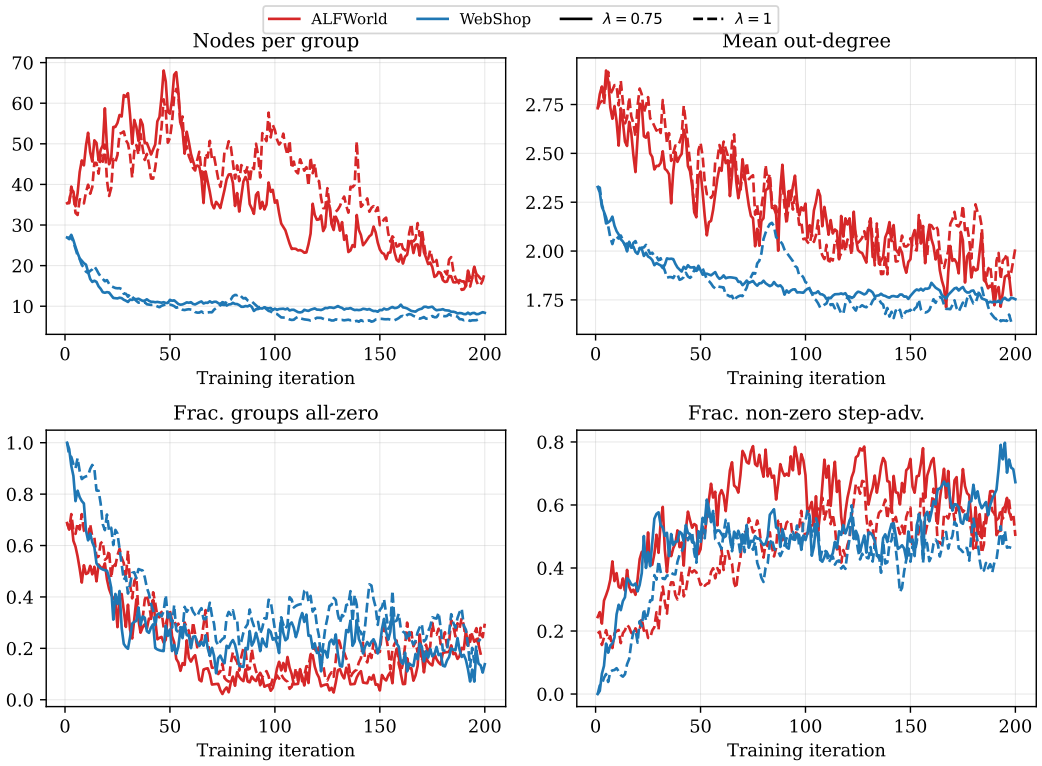


Figure 7: Graph and signal dynamics across training on Qwen2.5-1.5B-Instruct: active nodes per group, mean distinct-successor out-degree, the fraction of all-zero-signal groups, and the fraction of nonzero step advantages, for ALFWorld and WebShop under the default $\lambda = 0.75$ and the MC endpoint $\lambda = 1$. Solid lines are $\lambda = 0.75$ and dashed lines $\lambda = 1$; endpoint values are tabulated in Table 6.

transitions rather than degenerating into disjoint chains. The learning-dynamics summary tracks these signal differences: validation AUC is higher for $\lambda = 0.75$ in both environments (0.613 versus 0.494 on WebShop, 0.706 versus 0.625 on ALFWorld). On ALFWorld the two settings reach a similar validation endpoint (90.6 versus 92.2), but these last-logged values and AUCs span different horizons. They provide context for the learning traces, not a matched-window effect estimate.

The two traced settings, the default $\lambda = 0.75$ and the MC endpoint $\lambda = 1$, share prompts, rewards, initial weights, task and seed schedules, optimizer, and generation settings. The endpoint switches off graph bootstrapping, but the two policies evolve separately and need not produce the same batches. The intermediate weights are covered by the trace-sensitivity study.

Table 7: Learning-dynamics summary on Qwen2.5-1.5B-Instruct for the two traced λ settings. Train/Val success are recorded success rates (%); train/val AUC are the trapezoidal integrals of the corresponding success curves divided by the number of iterations (0–1 scale); episode length is the mean number of environment steps per episode. AUC horizons differ across these streams, so these summaries describe training dynamics rather than a matched-window performance comparison.

Environment	λ	Train succ. (%)	Val succ. (%)	Train AUC	Val AUC	Ep. length
WebShop	0.75	82.0	76.6	0.586	0.613	5.70
	1	68.0	62.5	0.489	0.494	5.07
ALFWorld	0.75	89.8	90.6	0.691	0.706	13.59
	1	87.5	92.2	0.596	0.625	15.38

The two mechanism cases are drawn from the same captured dumps as the offline study, in fixed task-ID order, taking one group per environment in which an interior state is visited by several trajectories with differing episode returns. No rollouts are added, and the quantities below come from the same graph primitives used for the offline predictors.

In the WebShop group at step 90 ($\lambda = 0.75$) all eight trajectories pass through a common interior state reached after the initial query. Seven episode MC returns cluster in 6.98–8.15 and one reaches 15.09; solving the empirical MDP of the group assigns that shared state a CE value of 9.95. The raw per-row MC returns at the state span 8.15 to 19.5 along the outlier branch, and the TD($\lambda = 0.75$) return with the CE value pulls the high-variance branch toward the graph-consistent value—$17.6 \to 13.9$ and $18.5 \to 16.3$—while leaving the well-supported median rows essentially unchanged ($9.5 \to 9.49$, $10.0 \to 10.0$). The pooled per-row mean moves only $10.38 \to 10.22$, so the estimator shrinks the outlier with little change in the mean. This illustrates within-group return shrinkage; one group cannot establish statistical unbiasedness or lower gradient variance.

The ALFWorld group at step 90 ($\lambda = 0.75$) shares a post-navigation state visited by all eight trajectories, with episode MC returns from 5.40 to 16.72 (one long successful trajectory) and a CE value of 8.82 at the state. The raw per-row MC returns there (5.69–17.6) are compressed toward the pooled value from both sides—the low branch lifted $5.69 \to 7.80$, the high branch shrunk $17.6 \to 13.98$—giving a per-row mean of 8.73 against the MC value of 8.67. The compression is stronger than on WebShop, matching the larger Real-CE margin ALFWorld shows in Table 4, which is what longer horizons and thinner per-anchor coverage predict. Both cases illustrate the mechanism on real groups and carry no win-rate claim.

C.6 Supplementary Settings: Visual Backbone and Group Size

These exploratory settings extend beyond the main protocol: a Sokoban visual-backbone transfer and a WebShop rollout-group-size study. The $n = 8$ row reuses the main result. The Sokoban summaries distinguish iteration 150 from the peak and plateau rather than treating them as matched comparisons.

Sokoban on a visual backbone. SGPO is trained on Sokoban [6×6] (Schrader, 2018) with Qwen2.5-VL-3B-Instruct (Bai et al., 2025), $\gamma = 0.95$, tensor-parallel 2, for 150 iterations with eight trajectories per group. This run fixes $\lambda = 0.5$ and uses per-anchor mean centering without standard-deviation normalization, and it depends on keying anchors by image so that distinct boards are not collapsed into one graph node. The Sokoban reward is the standard shaped signal (box-on-target minus a step penalty), so the reported score is the raw evaluation reward, roughly $[-1, 10]$ per episode rather than a 0–100 scale; success is the fraction of solved boards. The GiGPO baseline is taken from the reference implementation on the identical backbone and board size.

Table 8: Exploratory Sokoban [6×6] success (%) on Qwen2.5-VL-3B-Instruct.

Type	Method / summary	Success (%)
Prompting	Qwen2.5-VL	11.7
RL Training	GRPO	67.1
RL Training	GiGPO	76.9
RL Training	SGPO	73.4

WebShop rollout group size. With 16 task groups per iteration, varying n also changes the total number of rollout trajectories. Different logged endpoints and normalization conventions further prevent this sweep from isolating graph coverage at a fixed budget.

On WebShop we vary the number of trajectories per task group, $n \in \{2, 4, 6, 8\}$, holding history length 2, $\lambda = 0.75$, 16 task groups per iteration, and per-anchor mean centering fixed. Because the graph is built only from transitions a group actually visits, n directly controls how much cross-trajectory successor sharing the CE solver can use, which makes this a mechanism probe. The $n = 8$ row reuses the SGPO result from Table 1; training-budget and normalization differences are noted below.

The exploratory endpoints are ordered with group size: success climbs $49.2 \to 67.2 \to 75.8 \to 81.5$ and score $65.9 \to 75.8 \to 89.2 \to 89.6$. The curve is steep and noisy at $n = 2$—success oscillates in

Table 9: WebShop rollout-group-size study on Qwen2.5-1.5B-Instruct (history 2, $\lambda = 0.75$, 16 groups per iteration). Score is the WebShop task score and Succ. the success rate, both on a 0–100 scale. The $n = 8$ row is reproduced from Table 1 for comparison; budget and normalization differences prevent a controlled scaling comparison.

Group size n	Score	Succ.
$n = 2$	65.9	49.2
$n = 4$	75.8	67.2
$n = 6$	89.2	75.8
$n = 8$	89.6	**81.5**

a 5–49% band across training and the graph is nearly all singletons, so the CE bootstrap has almost no shared successors to exploit—and flattens between $n = 6$ and $n = 8$, where score is already saturated near 89 while success still gains about 6 points. This pattern is consistent with sparse successor sharing at small n, but it is not a controlled estimate of group-size scaling. The $n = 6$ run ends at iteration 140 rather than 160, and the budget and normalization differences noted above preclude isolating coverage. Appendix C.4 provides the separate fixed-policy test.

D Implementation and Reproducibility Details

For task-local graph construction, the graph is rebuilt independently for every task-conditioned rollout group. The eight trajectories in a group share the task, initial environment condition, and frozen behavior policy. Their counts define one task-local empirical anchor model; this locality alone does not establish the abstraction condition in Proposition 2. We do not pool anchor occurrences across tasks, environment seeds, or optimization batches. The node key is the exact environment-defined `anchor_obs` string. In ALFWorld it includes the admissible-command representation and progress information; in WebShop it is the current page observation without action history. Consequently, two rows are merged only when their keys are identical. We disable approximate similarity matching, making the coverage–aliasing trade-off explicit and reproducible.

Terminations, truncations, and padding are handled as follows. Natural termination and maximum-step truncation are both represented by a virtual absorbing node $\perp$ with $\widehat{V}(\perp) = 0$. No outgoing edge is added from $\perp$, and a terminal reward remains attached to the final real row. This convention separates the observed reward from the continuation value and also makes finite-horizon rollouts compatible with the same Bellman system. The training infrastructure can duplicate rows while padding a batch to a micro-batch size. Such duplicates retain the original (*trajectory*, *step*) identifier; SGPO removes them before estimating $(\widehat{r}, \widehat{P})$ and then copies the resulting advantage back to each duplicate. Without this step, padding could create artificial self-loops and inflate both transition counts and graph values.

The CE solution evaluates an empirical Markov reward process over anchors: it averages observed successors rather than maximizing over them. It is not automatically the value of the context-conditioned policy; that interpretation requires Proposition 2. Graph propagation can combine observed edges from different trajectories, and state aliasing can make the resulting graph walks infeasible from a particular history. Thus *no counterfactual transitions* means that no unobserved edge or new rollout is added, not that every graph walk is an executable trajectory. The TD recursion retains each sampled reward sequence and uses graph values only at its sampled successor anchors. Subsequent anchor normalization additionally couples rows in the same I_σ.

We now turn to the estimator output. After graph construction and the linear solve, the TD recursion is applied independently along each original trajectory. The estimator returns one scalar A_i^{step} for every sampled input row, in the original batch order. The outer optimizer therefore sees the same row-level interface as GiGPO: only the values inside the anchor-wise normalization change. This separation allows the graph estimator to be tested offline with held-out trajectories without changing the policy-update code, and makes the $\lambda = 1$ endpoint a direct implementation check rather than a separate training algorithm.

Increasing policy history length changes ψ, not the anchor map ϕ. For fixed logged rows, graph keys and grouping therefore stay unchanged; online, the changed policy can still alter visits, edges, and graph values. Such runs are context-length, not graph-key-refinement, ablations. Implementation checks cover natural termination, truncation, singleton anchors, repeated transitions, padding duplicates, and all-zero-reward groups.

For normalization, step residuals are centered and divided by the sample standard deviation within each anchor, using one consistent ϵ and degrees-of-freedom convention across methods. Singleton anchors output zero. Episode normalization and w_{step} are identical across the matched comparison.

Turning to numerics, the solver operates in floating point on the active-node system. We log condition numbers and fail loudly on non-finite inputs or solutions. The primary setting $\gamma = 0.95$ guarantees theoretical nonsingularity, but ill-conditioning remains a diagnostic. Sparse or iterative solving is unnecessary at the observed graph sizes.

We follow a result validation checklist. Before reporting results, we verify: (i) each metric is traceable to raw logs; (ii) evaluation task IDs and counts match; (iii) reported metrics and normalized AUC are regenerated from raw logs; (iv) confidence-interval code is versioned; (v) matched methods use the same step limit, prompt, and evaluation protocol; and (vi) every reported timing excludes warm-up under the same hardware load policy.