Group-Graph Policy Optimization for Long-Horizon Agentic Reinforcement Learning

Yunan Wang[1], **Minghui Song**[2], **Zihan Zhang**[2], **Shaohan Huang**[2], **Haizhen Huang**[2], **Furu Wei**[2], **Weiwei Deng**[2], **Feng Sun**[2], **Qi Zhang**[2]
[1]Peking University [2]Microsoft Corporation
yunanwang25@stu.pku.edu.cn

Abstract

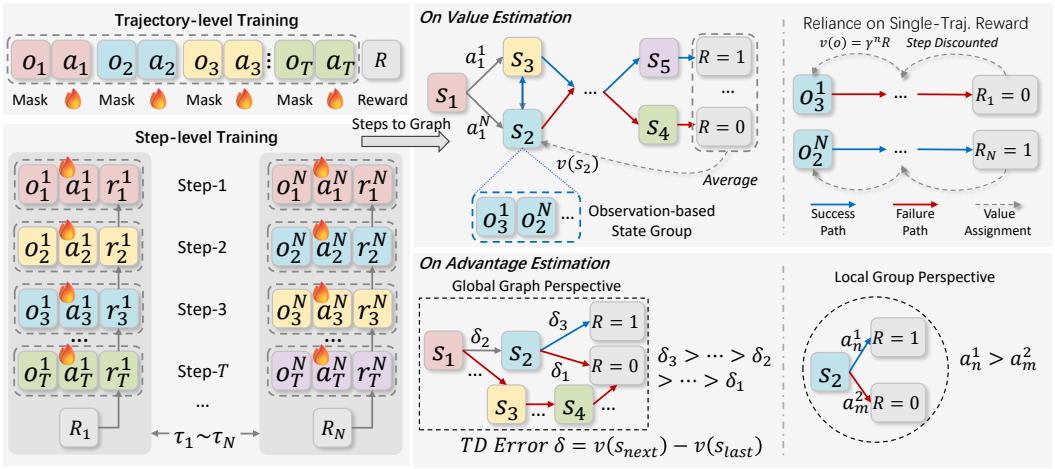
Group-based Reinforcement Learning (RL) has significantly enhanced Large Language Models (LLMs) in agentic scenarios. To achieve finer-grained policy updates, recent agentic RL frameworks have shifted from trajectory-level to step-level training. However, long-horizon agentic RL suffers from severe reward sparsity and delay, as feedback is often deferred for dozens of interaction steps. While existing step-level frameworks refine training granularity, their credit assignment remains coarse-grained and still treats agent exploration as isolated, linear trajectories. This oversimplified perspective ignores the inherent graph structure of state transitions, leading to high-variance state-value estimation and myopic, localized credit assignment. To overcome these critical bottlenecks, we propose Group-Graph Policy Optimization (G2PO), a novel group-based RL algorithm tailored for multi-turn agentic tasks. G2PO explicitly transforms linear interaction trajectories into a global state-transition graph. By aggregating identical observations across different trajectories, we introduce group-aggregation state-value estimation that reduces sampling variance and trajectory-dependent bias. Furthermore, we redefine agent actions as transitions between state nodes and propose an edge-centric advantage estimation strategy. By globally standardizing Temporal Difference (TD) errors across the entire graph, G2PO explicitly identifies and prioritizes critical transitions that drive absolute task progress. Extensive experiments on representative long-horizon benchmarks—WebShop, ALFWorld, and AppWorld—demonstrate that G2PO substantially outperforms state-of-the-art prompt-based and RL baselines, achieving remarkable success rate improvements of up to 22.2% over GRPO. Code available at https://github.com/Nala-YN/G2PO.

1 Introduction

The field of Large Language Models (LLMs) is witnessing a transformative leap from Chatbot to Copilot, and ultimately to *Agent* [14, 20, 26]. By interacting with real-world external environments (e.g., e-commerce websites [55], embodied physical simulations [43], and complex software ecosystems [46]), LLM agents are increasingly demonstrating the long-horizon decision-making capabilities to solve complex, real-world problems.

To enhance LLM reasoning, Reinforcement Learning (RL) has become a standard post-training pipeline for frontier models like GPT-5.2 [32], Gemini 3 Pro [16], and DeepSeek V3.2 [24]. Through algorithms such as PPO [39], GRPO [40], SAPO [15], and DAPO [58], RL has drastically improved model reasoning on math and coding tasks [6, 17, 36]. However, unlike single-turn reasoning, agentic RL typically requires the agent to interact with the environment for dozens of turns for a complete trajectory. Existing trajectory-level agentic RL frameworks, such as RAGEN [49] and Search-R1 [21], concatenate all environmental observations and policy actions into a single training sample. As the number of interaction turns increases, this trajectory-level training strategy leads to

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Traj. and Step-level Training (b) Comparison between G2PO and Step-level Training

Figure 1: (a) compares the training sample units between trajectory-level and step-level training. (b) contrasts G2PO (left) with existing step-level RL methods (right): while current approaches treat agent exploration as a collection of linear trajectories for credit assignment, G2PO tackles credit assignment from a global graph perspective.

an explosive growth in context length and extreme memory consumption. To address this, recent frameworks [8, 27, 47, 57] have shifted to step-level training. As shown in Figure 1(a), by treating each step's observation and action as an individual training sample, they resolve the context explosion problem and enable lightweight agentic RL. To further provide step-level advantage, GiGPO [13] adapts GRPO by calculating the relative advantage of actions within a group that shares the same environment observation.

Despite the refinement of training granularity, existing step-level RL methods still adopt coarse-grained credit assignment. These methods treat agent exploration as a collection of isolated, linear trajectories. This perspective fails to capture the true dynamics of multi-turn interactions, where different trajectories often encounter the same environmental states (e.g., the same webpage or physical configuration). These shared states naturally act as nodes to connect trajectories, with actions serving as edges that represent state transitions. Consequently, the underlying structure of a long-horizon task is inherently a state-transition graph. This structural oversight leads to two critical limitations in step-level training, as illustrated in Figure 1(b).

Limitation (i): High variance in state-value estimation. Due to the stochastic nature of sampling, different trajectories starting from the same state node often traverse distinct paths on the graph and terminate at different end-states. Evaluating a state's value based on only one of these possible outcomes is inherently unreliable. For example, a trajectory starting from a promising state might still fail due to subsequent errors, while one starting from a poor state might succeed through later corrections. Evaluating a state based on a single sample makes the estimation highly susceptible to chance and leads to high variance, as shown in Figure 4b.

Limitation (ii): Limited comparative scope of local group paradigms. Current methods (e.g., GiGPO) evaluate an action solely by comparing it to other available actions at the same local state. While this captures local preferences, it lacks the global perspective to resolve credit assignment across the entire graph horizon. Consequently, local comparisons fail to account for the global importance of an action, making it impossible to distinguish between a minor improvement in a trivial state and a critical breakthrough action that significantly advances the overall task progress.

To resolve these issues, we propose Group-Graph Policy Optimization (G2PO), a new group-based RL algorithm specifically designed for long-horizon agent tasks. We first explicitly reconstruct the sampled trajectories into a state-transition graph. Based on this graph, we introduce two novel designs: (i) *Group-aggregation state-value estimation*: Unlike prior studies that rely on a single trajectory's reward, we aggregate intermediate steps with identical or similar observations across all trajectories into a *state group*. The state value is then defined as the mean value of all states within this group. The key insight of this design is that the value of an environment state should be determined by its likelihood of leading to success, rather than a chance outcome from a single sample.

This aligns with the underlying philosophy of GRPO, which similarly leverages the empirical mean of all sampled outcomes as a baseline. (ii) *Edge-centric advantage estimation:* Within the state graph, agent actions function as directed edges representing state transitions. Having established robust value estimates for each state node, we can naturally evaluate an action based on the value increment between its source and destination nodes. In the context of sparse-reward agent tasks, this formulation aligns with the 1-step Temporal Difference (TD) error [44]. Crucially, rather than normalizing these advantages locally, we construct a global reference group comprising the TD errors of all transition edges across the entire graph. By globally standardizing these value increments, G2PO explicitly prioritizes *critical transitions* that drive the most substantial absolute progress toward the final goal.

We evaluate G2PO on three representative long-horizon agent benchmarks: WebShop, ALFWorld, and AppWorld. Experiments on Qwen-2.5-1.5B/7B/14B demonstrate that G2PO consistently outperforms existing prompt-based methods and RL training baselines. Notably, G2PO achieves success rate improvements of up to 14.4% and 22.2% over GRPO on WebShop and ALFWorld, respectively. The AppWorld results further validate its effectiveness in tasks with complex environmental observations. Notably, G2PO achieves these performance gains with almost no additional computational cost.

The main contributions of this paper are summarized as follows:

- We propose G2PO, a group-based reinforcement learning algorithm specifically designed for long-horizon agent tasks. By transforming linear interaction trajectories into an environment state group graph, G2PO enables fine-grained, step-wise training.
- We introduce group-aggregation state-value estimation to address the high variance in state-value estimation of existing group-based step-level RL methods.
- We redefine agent actions as transitions between nodes in the state group graph and design an edge-centric advantage estimation strategy that explicitly measures the absolute contribution of each step toward the global goal.
- We conduct extensive experiments across three demanding long-horizon agent benchmarks. G2PO consistently outperforms state-of-the-art prompt-based and RL baselines with almost no computational overhead.

2 Related work

LLM Agents. LLMs have evolved from passive text generators to autonomous agents capable of engaging in multi-turn interactions with external environments [19, 38, 48, 62]. Unlike single-turn QA tasks, agentic tasks (e.g., WebShop, AlfWorld) [11, 29, 53, 63] require the model to generate a sequence of actions over a long horizon to achieve a final goal. To solve agentic tasks, approaches such as ReAct [56], Toolformer [37], and Reflexion [41] demonstrate that interleaving reasoning traces with environment actions significantly enhances task performance. Agentic Reasoning [52] enhances LLM reasoning capabilities by integrating autonomous agentic tools into an efficient and structured problem-solving process. MetaGPT [18] improves multi-agent collaboration by encoding human standard operating procedures into a structured assembly-line paradigm for complex task-solving.

Reinforcement Learning for LLMs. RL for LLMs has evolved from initial preference alignment using PPO-based RLHF [3, 9, 33] and DPO [28, 35, 64] to enhancing complex reasoning capabilities through emergent group-based RL algorithms. By estimating advantages from multiple samples and eliminating the need for a critic network, methods such as RLOO and GRPO have enabled memory-efficient, large-scale training. However, these approaches predominantly rely on sparse outcome-based rewards, which suffer from the credit assignment problem by failing to distinguish the contributions of individual reasoning steps. While process reward models [4, 5, 23] offer step-level guidance, they often require prohibitive annotation costs and are prone to reward hacking. In contrast, G2PO leverages a state transition graph to derive dense, multi-granularity advantages directly from sampling statistics, effectively solving the credit assignment problem without the computational overhead of a learned critic or the annotation costs of process supervision.

Reinforcement Learning for LLM Agents. RL has emerged as a cornerstone for empowering LLM agents to adapt and thrive in dynamic, open-ended environments [25, 50, 51, 61]. While early efforts successfully adapted classical algorithms like DQN for text-based games [30, 31], subsequent research rapidly shifted toward policy gradient and value-based methods [2, 60]. Standard on-policy algorithms, such as PPO and AWR [34], proved effective in diverse interactive scenarios ranging from embodied tasks to complex card games. Despite these successes, efficiency and stability in

training remain critical challenges. CoSo [12] addresses this with entropy-based regularization, while LOOP [7] proposes a hybrid approach combining RLOO with PPO-style updates. Building on trajectory-level concepts, RAGEN [49] utilizes StarPO to optimize entire interaction trajectories. However, RAGEN's reliance on trajectory-level rewards prevents it from evaluating individual steps. GEPO [59] combines graph centrality and task similarity to estimate task-conditioned state criticality for credit assignment. While GiGPO uses anchor state grouping to compare steps at identical states, it still estimates rewards based on single-trajectory outcomes, which suffers from sampling bias. In contrast, our approach aggregates identical states across all sampled trajectories into a state group graph, achieving fine-grained reward estimation with significantly lower variance.

3 Preliminary

3.1 Problem Setup

We consider a general scenario where an LLM agent solves a given task $x \in X$ through multi-turn, long-horizon interactions with an environment. At each time step $t = 1, 2, 3 \ldots, T$, the agent receives an observation o_t (where $o_1 = x$) from the environment and produces a new action a_t. The environment receives a_t and then returns the subsequent observation o_{t+1}. This cycle yields a complete trajectory $\tau = (o_1, a_1, \ldots, o_T, a_T)$. The observation here incorporates a fixed number of historical steps. The agent's policy can be formulated as $\pi_\theta(a_i|o_i)$, where θ represents the model parameters. Only when the task completed or the interaction threshold reached, the environment returns a scalar reward R. Consequently, the agent receives no immediate rewards during intermediate steps, which makes scoring the agent's intermediate actions challenging.

3.2 Group Relative Policy Optimization

Recently, Group Relative Policy Optimization (GRPO) has served as a popular reinforcement learning algorithm that eliminates the computational burden of maintaining a separate value function critic. Instead, GRPO estimates the baseline using the mean reward of a group of sampled outputs. For each query q, GRPO generates a group of G outputs $\{y_1, y_2, \ldots, y_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$. The advantage A_i for the i-th output is computed by normalizing its reward R_i against the group statistics: $A_i = \frac{R_i - \mu}{\sigma}$, where μ and σ represent the mean and standard deviation of the rewards within the group, respectively. The objective function maximizes the clipped surrogate objective while incorporating a KL-divergence penalty to ensure training stability:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D} \\ \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}} \left[\frac{1}{G}\sum_{i=1}^{G} \left(\min\left(\rho_i A_i, \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon) A_i\right) - \beta \mathbb{D}_{\text{KL}}(\pi_\theta(\cdot|q)||\pi_{\text{ref}}(\cdot|q)) \right) \right], \quad (1)$$

where $\rho_i = \frac{\pi_\theta(y_i|q)}{\pi_{\theta_{\text{old}}}(y_i|q)}$ denotes the importance sampling ratio, ϵ is the clipping parameter, and β controls the strength of the regularization towards the reference policy π_{ref}.

4 Training LLM Agents with G2PO

In this section, we propose G2PO to provide fine-grained credit assignment for multi-turn agentic RL from the perspective of a state group graph, as illustrated in Figure 2. We first introduce the state graph construction strategy. We then present the state-value estimation method designed to mitigate sampling variance and finally detail the advantage calculation process.

4.1 State Group Graph Construction

In multi-turn agentic RL, the environment returns an observation in each step. The observation allows us to infer the corresponding environmental state and group the states across different sampled trajectories. For instance, in e-commerce scenario, identical web page content at different intermediate steps can be considered to belong to the same state group. Transitions between these state group nodes occur via the agent's actions, thereby forming the edges.

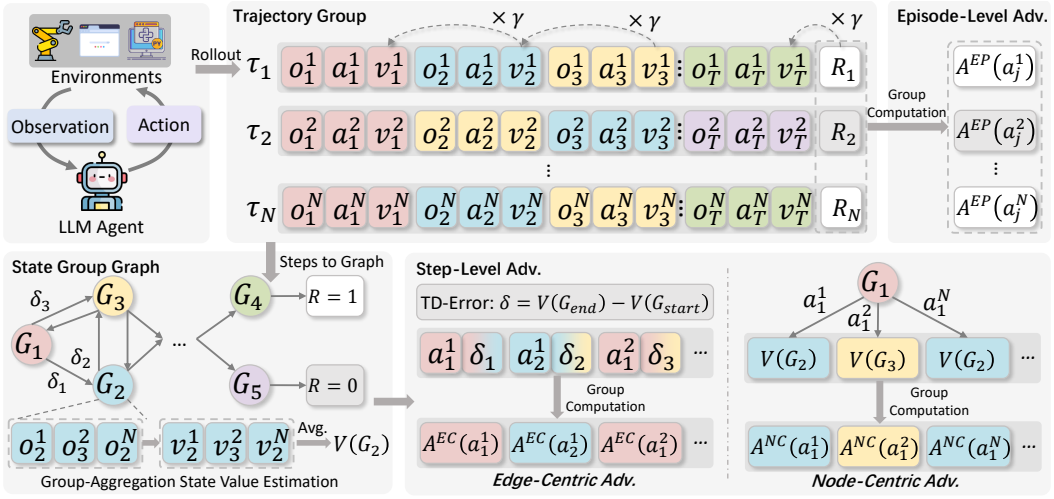


Figure 2: Overview of G2PO. For the sampled trajectories, we first construct a state group graph by aggregating steps with similar observations (same color) into the same state groups. We then define the state-group value as the average value of its constituent observations. Based on this graph, we provide step-level advantages from the edge and node perspectives, respectively.

Formally, for a given task x, we first sample N complete trajectories $\{\tau_i\}_{i=1}^{N}$ using the policy $\pi_{\theta_{old}}$, where $\tau_i = \{o_1^i, a_1^i, o_2^i, a_2^i, ..., o_T^i, a_T^i\}$ and all trajectories satisfy $o_1^i = x$. Let $\mathcal{O} = \{o_j^i\}_{i=1,j=1}^{N,T}$ be the set of all intermediate observations. We then partition $\mathcal{O}$ into G disjoint state groups $\{\mathcal{G}_k\}_{k=1}^{G}$ by clustering identical observations. Specifically, each group is defined as $\mathcal{G}_k = \{o_j^i \in \mathcal{O} \mid o_j^i = \bar{o}_k\}$, where $\bar{o}_k$ represents the unique observation characterizing the k-th state group. By treating these state groups as nodes and agent actions as edges, we construct a state transition graph $\{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{\mathcal{G}_k\}_{k=1}^{G}$ and $\mathcal{E} = \{(\mathcal{G}_s, a, \mathcal{G}_t) \mid \exists i, j \text{ s.t. } o_j^i \in \mathcal{G}_s, a_j^i = a, o_{j+1}^i \in \mathcal{G}_t\}$.

4.2 Group-Aggregation State-Value Estimation

After constructing the state graph, different from prior studies that directly use the reward of a single trajectory to calculate the value of intermediate states, we average step reward of all intermediate steps within the state node as the state value. Our motivation is twofold:

- Variance reduction: Due to sampling randomness, identical states in different trajectories may lead to different outcomes. Reliance on raw reward of a single sampling for value calculation introduces significant variance.
- Improved temporal credit assignment: In multi-turn RL, a good action may fail due to later errors, while a mediocre one succeeds by luck. Group aggregation averages out this future-step noise, ensuring more reliable and stable state evaluation.

By averaging across different attempts passing through the same state node, we estimate the intrinsic expected value of the state, effectively mitigating value estimation variance, as formally proven in Appendix B.1. Specifically, we first independently calculate the state value for steps in each trajectory using discounted returns:

$$v_j^i = \gamma^{T-j+1} R_i, \tag{2}$$

where $\gamma \leq 1$ is the discount factor and R_i is the outcome reward of τ_i. Subsequently, for each state group $\mathcal{G}_k$, we define its value as the average of the values of all constituent steps:

$$V(\mathcal{G}_k) = \frac{1}{|\mathcal{G}_k|} \sum_{o_j^i \in \mathcal{G}_k} v_j^i. \tag{3}$$

4.3 Multi-Granularity Advantage Estimation

In agentic RL, GRPO typically calculates advantages across trajectories based solely on the outcome reward. Steps within the corresponding trajectory share this advantage. This coarse approach can mislead the optimization, where good actions receive negative advantages due to subsequent failures,

or mediocre actions receive positive advantages due to subsequent corrections. To address this, we leverage the state group graph to synthesize three granularities of advantage to provide dense and multi-granularity feedback.

Node-Centric Advantage Estimation. When the agent observes a specific environmental state $\bar{o}_k$, it may select from various actions leading to different subsequent states. These actions, originating from the same node, constitute a node-centric group. Formally, for any state group $\mathcal{G}_k$, we collect the values of all next-states resulting from transitions out of $\mathcal{G}_k$ to form a local reference set $\Delta_k = \{V(\mathcal{G}) \mid o_m^n \in \mathcal{G}_k,\ o_{m+1}^n \in \mathcal{G}\}$. For any observation o_j^i in $\mathcal{G}_k$, the node-centric advantage of corresponding action $a_j^i$is calculated as:

$$A^{NC}(a_j^i) = \frac{V(\mathcal{G}_{k'}) - \text{mean}(\Delta_k)}{\text{std}(\Delta_k)}, \tag{4}$$

where next observation $o_{j+1}^i \in \mathcal{G}_k'$.

Edge-Centric Advantage Estimation. Within the state group graph, executing action a_j^i transitions the environments from the current state group $\mathcal{G}_k$ ($o_j^i \in \mathcal{G}_k$) to the subsequent state group $\mathcal{G}_{k'}$ ($o_{j+1}^i \in \mathcal{G}_{k'}$). We quantify the quality of this transition by measuring its intrinsic value gain. In typical long-horizon agent tasks with sparse terminal rewards and an undiscounted horizon (i.e., $\gamma = 1$), this value gain structurally corresponds to the 1-step TD error [44], defined as:

$$\delta_j^i = V(\mathcal{G}_{k'}) - V(\mathcal{G}_k). \tag{5}$$

Standard node-centric methods normalize advantages locally among actions sharing the same state. However, local normalization obscures the absolute significance of a transition. A marginally better action in a trivial state might improperly receive the same advantage score as a critical breakthrough action. To capture the global significance of each transition, we propose treating the TD errors of all intermediate steps across all sampled trajectories as a global reference set. The edge-centric advantage is then computed via global standardization:

$$A^{EC}(a_j^i) = \frac{\delta_j^i - \text{mean}(\{\delta_j^i\}_{i=1,j=1}^{N,T})}{\text{std}(\{\delta_j^i\}_{i=1,j=1}^{N,T})}. \tag{6}$$

The theoretical insight of this edge-centric formulation lies in its robust approach to the temporal credit assignment problem in sparse-reward RL. By globally standardizing TD errors across the entire state-transition graph, A^{EC} explicitly identifies and prioritizes *critical transitions*—actions that yield a substantial leap toward task completion. Consequently, an impactful action receives proportional positive reinforcement even if its specific trajectory ultimately fails due to subsequent exploratory errors. Conversely, trivial actions in successful trajectories are naturally penalized or down-weighted, effectively mitigating the noise inherent in long-horizon exploration. Meanwhile, this approach can reduce variance of advantage estimation compared to GRPO when the group size exceeds 1, as formally proven in Appendix B.2.

Episode-Level Advantage Estimation. Although A^{NC} and A^{EC} provide dense and fine-grained feedback, it is crucial to ensure the learning process does not deviate from the ultimate task objective. Therefore, we retain an episode-level advantage similar to the original GRPO:

$$A^{EP}(a_j^i) = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{N})}{\text{std}(\{R_i\}_{i=1}^{N})}, \tag{7}$$

where R_i is the final reward for trajectory τ_i.

Finally, we merge the episode-level and step-level advantage to compute the overall advantage for each step:

$$A_j^i = A^{EP}(a_j^i) + w \cdot (A^{NC}(a_j^i) + A^{EC}(a_j^i)), \tag{8}$$

where w is a static weight used to scale the contribution of the step-level advantage.

Following the GRPO framework, the overall objective function of G2PO is formulated as:

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{x\sim p(X) \\ \{\tau_i\}_{i=1}^{N}\sim\pi_{\theta_{\text{old}}}}} \left[\frac{1}{NT} \sum_{i=1}^{N} \sum_{j=1}^{T} \left(\min\left(\rho_j^i A_j^i, \text{clip}\left(\rho_j^i, 1-\epsilon, 1+\epsilon\right) A_j^i\right) - \beta D_{\text{KL}}\left(\pi_\theta(\cdot|o_j^i) \| \pi_{\text{ref}}(\cdot|o_j^i)\right)\right)\right] \tag{9}$$

where $\rho_j^i = \frac{\pi_\theta(a_j^i|o_j^i)}{\pi_{\theta_{old}}(a_j^i|o_j^i)}$ denotes the importance sampling ratio. ϵ is the clipping parameter to constrain policy updates, and β is the coefficient for the KL-divergence penalty term D_{KL}.

5 Experiment

5.1 Experimental Setting

Benchmarks. We conduct experiments on three challenging agentic benchmarks across different scenarios: WebShop [55] for e-commerce, ALFWorld [43] for embodied physical environments, and AppWorld [46] for complex software ecosystems. These benchmarks comprehensively assess the agents' long-horizon decision-making abilities in complex environments.

Baselines. We comprehensively compare G2PO against three categories of competitive baselines: (1) Frontier off-the-shelf models, representing the upper bound of direct-inference capabilities, including Gemini-2.5-Pro [10], DeepSeek-V3.2 [24], and Qwen3.5-397B [54]; (2) Prompting agents, that leverage inference-time compute, including ReAct [56] and Reflexion [41]; (3) RL training methods, including PPO [39] as the standard actor-critic baseline, RLOO [1] which bypasses the critic model via a leave-one-out advantage baseline, GRPO [40] which normalizes rewards within sampled groups to reduce memory overhead, and GiGPO [13] which introduces sub-grouping structures for fine-grained advantage estimation.

Implementation details. We primarily use Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct [45] as base models, while adopting Qwen2.5-14B-Instruct for AppWorld due to its high difficulty. All RL methods share the same hyperparameters and use identical prompts for rollouts, where <think></think> encapsulates the reasoning process and <action></action> specifies the executed action. We apply an invalid action penalty with a coefficient of 0.1 and set the learning rate to 1e-6. We use vLLM [22] as the inference engine. For group-based methods, the number of rollouts per task N is set to 8 in all main experiments. Further implementation details can be found in Appendix E.

State grouping. All main-paper experiments group observations by exact string equality: webpage content (WebShop), local environment descriptions (ALFWorld), and Python API returns or errors (AppWorld). Exact matching is an implementation choice rather than an algorithmic requirement. G2PO only requires an observation-to-node mapping; canonicalization or lexical/semantic similarity can replace exact matching without changing value estimation or optimization. Appendix G details a noisy ALFWorld experiment validating this flexibility.

5.2 Experimental Results

Performance on WebShop and ALFWorld. As shown in Table 1, even frontier LLMs fall short on these two benchmarks. For example, Qwen3.5-397B yields only a 17.2% success rate on WebShop. Moreover, RL training methods significantly outperform prompt-based methods, highlighting their critical role in internalizing agent capabilities into the model. Among these RL methods, G2PO consistently surpasses all baselines at both 1.5B and 7B scales. In particular, G2PO achieves improvements with a substantial margin of up to 22% and 14% over the GRPO baseline in WebShop and ALFWorld, respectively. Furthermore, compared to GiGPO which relies on comparison within local groups, G2PO still performs better, demonstrating the importance of edge-centric advantage which offers global signals.

Performance on AppWorld. As shown in Table 2, G2PO outperforms all RL training methods, proving its effectiveness and strong generalizability even on a more challenging benchmark like AppWorld. In AppWorld, agents interact with environments via Python API calls. This requires the agent to infer the app's underlying data state from specific API returns and effectively execute subsequent actions based on this state. Despite such complex observations, G2PO can still construct state transition graphs and provide fine-grained credit assignment to distinguish valid from invalid actions. This highlights G2PO's robust generalizability across diverse observation modalities.

Table 2: Results on AppWorld.

Type	Method	Succ.	Score
Qwen2.5-14B-Instruct			
Prompting	Qwen2.5	0.0	0.0
Prompting	ReAct	10.5	8.0
RL Training	PPO (with critic)	19.1	14.7
RL Training	RLOO	24.8	19.5
RL Training	GiGPO	25.7	19.2
RL Training	GRPO	24.8	20.7
RL Training	G2PO	**27.6**	**21.7**

Table 1: Results on WebShop and ALFWorld. We report the success rate (%) for each sub-task and overall results for ALFWorld. We report the score and success rate (%) for WebShop. Results are averaged over 3 random seeds. The best results are highlighted in **bold**.

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Off-the-Shelf Model										
Prompting	Qwen3.5-397B	96.6	72.7	70.4	0.0	70.4	85.7	71.9	27.6	17.2
Prompting	DeepSeek-V3.2	79.3	54.6	48.2	46.2	29.6	57.1	53.1	31.6	18.0
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)	$64.8_{\pm 3.5}$	$40.5_{\pm 6.9}$	$57.1_{\pm 4.9}$	$60.6_{\pm 6.6}$	$46.4_{\pm 4.0}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$	$73.8_{\pm 3.0}$	$51.5_{\pm 2.9}$
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	GiGPO	$94.4_{\pm 5.9}$	$67.5_{\pm 4.6}$	$94.8_{\pm 3.8}$	$94.4_{\pm 7.8}$	$79.8_{\pm 4.7}$	$76.4_{\pm 5.4}$	$86.7_{\pm 1.7}$	$83.5_{\pm 1.8}$	$67.4_{\pm 4.5}$
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	**G2PO**	$\mathbf{97.2}_{\pm 3.9}$	$\mathbf{96.3}_{\pm 5.2}$	$\mathbf{95.1}_{\pm 1.8}$	$\mathbf{94.9}_{\pm 3.6}$	$\mathbf{97.1}_{\pm 2.0}$	$\mathbf{91.4}_{\pm 4.1}$	$\mathbf{95.0}_{\pm 0.8}$	$\mathbf{85.1}_{\pm 1.4}$	$\mathbf{71.2}_{\pm 2.6}$
RL Training	Δ *vs GRPO*	*+11.9*	*+42.6*	*+10.6*	*+16.7*	*+37.4*	*+37.9*	*+22.2*	*+9.3*	*+14.4*
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)	$92.3_{\pm 4.0}$	$64.0_{\pm 8.4}$	$92.5_{\pm 2.4}$	$89.5_{\pm 7.0}$	$80.3_{\pm 2.0}$	$68.8_{\pm 8.3}$	$80.4_{\pm 2.7}$	$81.4_{\pm 3.1}$	$68.7_{\pm 5.1}$
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	GiGPO	$97.7_{\pm 1.6}$	$82.7_{\pm 7.9}$	$\mathbf{98.8}_{\pm 1.6}$	$83.7_{\pm 7.2}$	$89.3_{\pm 8.2}$	$79.2_{\pm 6.6}$	$90.8_{\pm 1.3}$	$86.2_{\pm 2.6}$	$75.2_{\pm 3.8}$
RL Training	GRPO	$90.8_{\pm 5.1}$	$66.1_{\pm 6.7}$	$89.3_{\pm 5.4}$	$74.7_{\pm 6.9}$	$72.5_{\pm 5.4}$	$64.7_{\pm 7.3}$	$77.6_{\pm 5.2}$	$79.3_{\pm 2.8}$	$66.1_{\pm 3.7}$
RL Training	**G2PO**	$\mathbf{98.7}_{\pm 2.3}$	$\mathbf{90.6}_{\pm 11.5}$	$97.6_{\pm 2.1}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{98.8}_{\pm 2.1}$	$\mathbf{93.7}_{\pm 8.4}$	$\mathbf{96.9}_{\pm 1.3}$	$\mathbf{89.8}_{\pm 1.7}$	$\mathbf{78.3}_{\pm 0.6}$
RL Training	Δ *vs GRPO*	*+7.9*	*+24.5*	*+8.3*	*+25.3*	*+26.3*	*+29*	*+19.3*	*+10.5*	*+12.2*

5.3 Ablation Study

In this part, we conduct ablation studies to validate the key components of G2PO. We start with a baseline Episode-level Advantage (A^{EP}) and incrementally introduce Node-Centric Advantage (A^{NC}), Group-Aggregation (GA) state-value estimation, and Edge-Centric Advantage (A^{EC}) to evaluate their individual contributions. We train on Qwen2.5-1.5B-Instruct for ablation results.

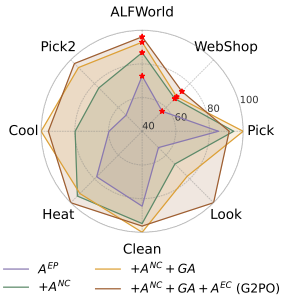
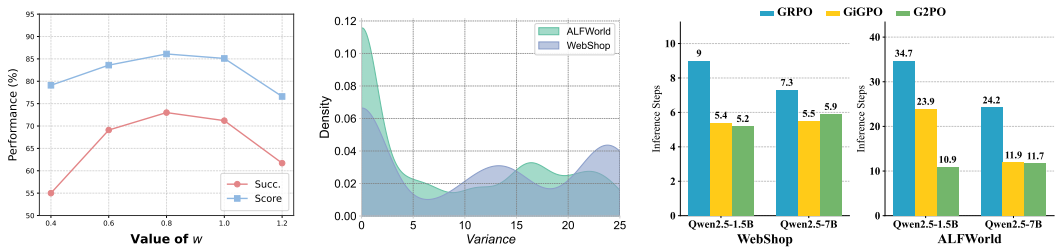


Figure 3: Ablation results. The y-axis shows success rate (%).

As illustrated in Figure 3, integrating the group-aggregation state-value estimation yields significant performance gains on both WebShop and ALFWorld. This improvement highlights its vital role in reducing the variance of state-value estimation by averaging the outcomes of identical states encountered across multiple trajectories. Furthermore, incorporating edge-centric advantage estimation outperforms relying solely on the node-centric approach. This confirms that the edge-centric method successfully captures the global, absolute contribution of actions, which local comparisons within a state group fail to reflect. Overall, the ablation results demonstrate that every component of G2PO is indispensable for providing accurate and fine-grained advantage. Appendix I further examines whether edge-centric advantages assign greater credit to critical actions and correlate with action importance.

5.4 Analysis

Hyperparameter Analysis. Here we analyze the parameter w in Eq. 8, which balances step-level and episode-level advantages. We conduct experiments using Qwen2.5-1.5B on WebShop to evaluate its impact. As shown in Figure 4a, the model's performance first increases and then decreases as w grows, achieving the highest success rate and score at $w = 0.8$. This highlights the parameter's critical role: when w is too low, the algorithm degenerates to GRPO and fails to provide fine-grained advantages for individual steps; when w is too high, the algorithm becomes overly short-sighted, causing the optimization to easily deviate from the ultimate task objective.



(a) (b) (c)

Figure 4: Results for analysis. Figure 4a: results under different values of parameter w; Figure 4b: variance distribution of state-value estimation with step-level RL methods, highlighting the necessity of group-aggregation mechanism to reduce variance; Figure 4c: the number of inference steps required to finish tasks. G2PO demonstrates substantial advantages over existing methods in inference efficiency.

State-Value Variance Analysis. To illustrate the necessity of the group-aggregation mechanism in G2PO, we present the variance statistics of value estimation with current step-level RL methods. We distinguish states based on environmental observations and collect the final task reward derived from a state as its state value. As illustrated in Figure 4b, across both benchmarks and 1.5B/7B model scales, a significant portion of states exhibits a non-zero value variance. This implies that starting from the same state, the agent often reaches different outcome states with varying rewards, resulting in inconsistent state values for identical states across different trajectories. To resolve this high-variance issue, our group-aggregation mechanism averages the diverse outcomes stemming from identical states across trajectories, assigning a robust and consistent value to each state node.

Task Efficiency Analysis. As shown in Figure 4c, we collected the number of environment interaction turns required by the trained models to complete tasks. It is evident that, compared to GRPO, models trained with G2PO complete tasks in fewer interaction turns on both WebShop and ALFWorld. In most cases, they also require fewer turns than GiGPO, indicating that G2PO enhances the model's ability to make efficient decisions. Fewer interaction turns inherently translate to lower model inference costs and environment API call expenses. Consequently, G2PO demonstrates substantial advantages over existing methods in both task accuracy and cost efficiency.

5.5 Computational Budget

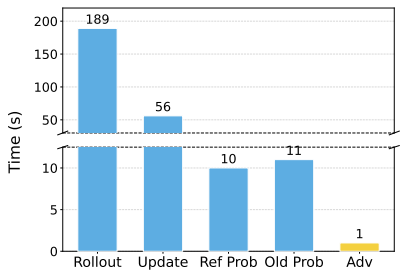


Figure 5: Breakdown of time consumption per training step. G2PO introduces only 1s of additional time overhead when computing advantages.

In this section, we demonstrate the efficiency of G2PO by analyzing the time consumption per training step. We use Qwen2.5-1.5B-Instruct as the base model and train it on the ALFWorld benchmark.

As illustrated in Figure 5, a typical training step involves a multi-turn sampling process, calculating the output probabilities of the sampled responses under both the reference and old policies, estimating advantages, and updating the policy. G2PO operates specifically within the advantage estimation stage. We can see from the result that G2PO introduces only 1 second of additional overhead during advantage calculation, successfully preserving the high efficiency of GRPO. It is mainly due to its GPU-free design. Specifically, to estimate advantages, G2PO constructs a state graph based on environmental observations, performs value estimation via group aggregation, and computes advantages at three different granularities. As this entire pipeline relies exclusively on the CPU, it occupies a mere 0.4% of the total RL training time, introducing virtually no extra computational cost.

6 Conclusion

In this paper, we propose G2PO, a novel group-based reinforcement learning framework for long-horizon agent tasks. By restructuring isolated interaction trajectories into a unified state-transition graph, G2PO unlocks two key innovations: (i) *group-aggregation state-value estimation*, which mitigates the high variance of single-trajectory outcomes, and (ii) *edge-centric advantage estimation*, which evaluates the intrinsic value gain of each transition to globally prioritize critical breakthroughs. Experiments demonstrate that G2PO achieves substantial performance gains over state-of-the-art RL baselines with almost no additional computational overhead.

Acknowledgments and Disclosure of Funding

Funding. This work was supported by computing resources provided by Microsoft.

Competing interests. The authors declare no competing financial interests.

References

[1] 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)*, pages 12248–12267, 2024.

[2] 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)*, pages 3557–3565, 2019.

[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[4] Lang Cao, Renhong Chen, Yingtian Zou, Chao Peng, Huacong Xu, Yuxian Wang, Wu Ning, Qian Chen, Mofan Peng, Zijie Chen, et al. More bang for the buck: Process reward modeling with entropy-driven uncertainty. *arXiv preprint arXiv:2503.22233*, 2025.

[5] Qi Cao, Ruiyi Wang, Ruiyi Zhang, Sai Ashish Somayajula, and Pengtao Xie. Dreamprm: Domain-reweighted process reward model for multimodal reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[6] Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, et al. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*, 2026.

[7] 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.

[8] Wentse Chen, Jiayu Chen, Hao Zhu, and Jeff Schneider. Context-lite multi-turn reinforcement learning for llm agents. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025.

[9] 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.

[10] 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.

[11] 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.

[12] Lang Feng, Weihao Tan, Zhiyi Lyu, Longtao Zheng, Haiyang Xu, Ming Yan, Fei Huang, and Bo An. Towards efficient online tuning of vlm agents via counterfactual soft reinforcement learning. In *International Conference on Machine Learning*, pages 16884–16903. PMLR, 2025.

[13] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[14] Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

[15] Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

[16] Google. A new era of intelligence with gemini 3, 2025.

[17] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[18] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*, 2023.

[19] Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jian-Guang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 496–507, 2025.

[20] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*, 2024.

[21] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

[22] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[23] Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *The Thirteenth International Conference on Learning Representations*.

[24] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[25] Shuo Liu, Tianle Chen, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. *arXiv preprint arXiv:2508.04652*, 2025.

[26] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

[27] Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025.

[28] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[29] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

[30] 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.

[31] 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*, pages 1–11, 2015.

[32] OpenAI. Introducing gpt-5.2, 2025.

[33] 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.

[34] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[35] 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.

[36] ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025.

[37] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

[38] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043, 2025.

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

[40] 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.

[41] 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.

[42] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[43] 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.

[44] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction second edition. *Adaptive computation and machine learning: The MIT Press, Cambridge, MA and London*, 2018.

[45] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[46] Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076, 2024.

[47] Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025.

[48] Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024.

[49] 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.

[50] Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. *arXiv preprint arXiv:2505.15107*, 2025.

[51] Taylor Webb, Shanka Subhra Mondal, and Ida Momennejad. A brain-inspired agentic architecture to improve planning with llms. *Nature Communications*, 16(1):8633, 2025.

[52] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research.

[53] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

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

[55] 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, 2022.

[56] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

[57] Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

[58] 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. *arXiv preprint arXiv:2503.14476*, 2025.

[59] Jiazhen Yuan, Zhike Gong, Jinquan Hang, Zhengbiao Bai, and Wei Zhao. Graph-enhanced policy optimization in llm agent training. *arXiv preprint arXiv:2510.26270*, 2025.

[60] Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *international conference on machine learning*, pages 12333–12344. PMLR, 2021.

[61] Hanchen Zhang, Xiao Liu, Bowen Lv, Xueqiao Sun, Bohao Jing, Iat Long Iong, Zhenyu Hou, Zehan Qi, Hanyu Lai, Yifan Xu, et al. Agentrl: Scaling agentic reinforcement learning with a multi-turn, multi-task framework. *arXiv preprint arXiv:2510.04206*, 2025.

[62] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62138–62160, 2024.

[63] 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 *The Twelfth International Conference on Learning Representations*.

[64] Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. Wpo: Enhancing rlhf with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8328–8340, 2024.

A Limitations

One limitation of our paper is that we only implement our method on the verl-agent [13] framework, considering that it provides lightweight step-level RL and flexible credit assignment control. Experiments on more agentic RL frameworks will further demonstrate the usefulness of G2PO. Another direction worthy of exploration is extending G2PO to more rollout settings, such as tree sampling and asynchronous sampling.

B Variance Reduction Analysis

In this section, we provide theoretical analysis and proof for the reduction of variance in value estimation through group-aggregation state-value estimation, as well as the reduction of variance in advantage estimation through edge-centric advantage estimation compared to trajectory-level credit assignment.

B.1 Value Estimation Variance Reduction

In long-horizon agent tasks with sparse terminal rewards and an undiscounted horizon ($\gamma = 1$), intermediate steps yield zero immediate rewards (i.e., $r_t = 0$ for $t < T$). Consequently, the cumulative empirical return from any intermediate step j precisely collapses to the final terminal reward R_i:

$$v_j^i = \sum_{t=j}^{T} \gamma^{t-j} r_t = 0 + 0 + \cdots + R_i = R_i.$$

Let $Var(R_i) = \sigma^2$. We have:

- **Variance of Single-Trajectory Estimation:** In previous studies utilizing single-trajectory returns, the variance of the estimate is:

$$\text{Var}(v_j^i) = \text{Var}(R_i) = \sigma^2.$$

- **Variance of G2PO (State Group Aggregation):** For convenience, we assume that the sampling across different trajectories $\{\tau_i\}$ is independent and each intermediate step within $\mathcal{G}_k$ is sampled from a different trajectory. Then the variance of our aggregated estimate is:

$$\text{Var}(V(\mathcal{G}_k)) = \text{Var}\left(\frac{1}{|\mathcal{G}_k|} \sum_{o_j^i \in \mathcal{G}_k} v_j^i\right) = \frac{1}{|\mathcal{G}_k|^2} \sum_{o_j^i \in \mathcal{G}_k} \text{Var}(R_i) = \frac{1}{|\mathcal{G}_k|^2} \cdot |\mathcal{G}_k| \cdot \sigma^2 = \frac{\sigma^2}{|\mathcal{G}_k|}.$$

Consequently, when the state group size $|\mathcal{G}_k| > 1$, the variance of the value estimation decreases linearly relative to the single-trajectory estimate.

B.2 Advantage Estimation Variance Reduction

Here, we follow the assumptions used in the proof of the previous subsection.

- **Variance of Traditional Trajectory-Level Advantage:** The trajectory-level advantage can be essentially viewed as $A^{traj}(a_j^i) = R_i - b$ (where b is a scalar baseline). The variance of this estimation is entirely determined by the single-sample trajectory outcome:

  $$\text{Var}(A^{traj}(a_j^i)) = \text{Var}(R_i) = \sigma^2.$$

  This implies that trajectory-level advantage estimation is heavily influenced by the outcomes of subsequent steps, thereby failing to accurately assess the intrinsic expected advantage of the action itself.
- **Variance Reduction in Edge-Centric Advantage:** In contrast, G2PO evaluates the action via the edge-centric advantage, which relies on the 1-step Temporal Difference (TD) error between the target and source state groups:

  $$\delta_j^i = V(\mathcal{G}_{k'}) - V(\mathcal{G}_k).$$

  According to the fundamental properties of variance, this expands to:

  $$\text{Var}(\delta_j^i) = \text{Var}(V(\mathcal{G}_{k'}) - V(\mathcal{G}_k)) = \text{Var}(V(\mathcal{G}_{k'})) + \text{Var}(V(\mathcal{G}_k)) - 2\text{Cov}(V(\mathcal{G}_{k'}), V(\mathcal{G}_k)).$$

  Since trajectories passing through the target node $\mathcal{G}_{k'}$ inherently share historical context or overlap with those passing through the source node $\mathcal{G}_k$, their value estimates are positively correlated, i.e., $\text{Cov}(V(\mathcal{G}_{k'}), V(\mathcal{G}_k)) \geq 0$. Combining this with the variance of the group-aggregated state value derived previously, we obtain an upper bound for the variance of the edge-centric advantage:

  $$\text{Var}(\delta_j^i) \leq \text{Var}(V(\mathcal{G}_{k'})) + \text{Var}(V(\mathcal{G}_k)) = \frac{\sigma^2}{|\mathcal{G}_{k'}|} + \frac{\sigma^2}{|\mathcal{G}_k|}.$$

  Thus, when $|\mathcal{G}_k| > 1$ and $|\mathcal{G}_{k'}| > 1$, we have

  $$\text{Var}(\delta_j^i) \leq \sigma^2.$$

  Consequently, the Edge-Centric Advantage formulation guarantees that, under these assumptions, the estimation variance is upper-bounded by the variance of trajectory-level advantage estimation.

C Pseudo Code

Algorithm 1 summarizes the full G2PO training procedure. Compared to vanilla GRPO, we highlight the additional parts introduced by G2PO in italics. In particular, constructing the state group graph and estimating state values through the group-aggregation mechanism mitigate sampling variance and trajectory-dependent bias. Furthermore, computing edge-centric and node-centric advantages provides multi-granularity feedback, which are adaptively integrated with episode-level advantages using confidence-aware weights. As such, G2PO provides dense and fine-grained credit assignment while ensuring the learning process aligns with the ultimate task objective.

D Group Size Analysis

Here we present the dynamics of the average state group size during training and the distribution of group sizes. As illustrated in Figure 6, the average group size for both WebShop and ALFWorld is around 5. This indicates that many environmental observations are identical during training but may lead to different outcomes, highlighting the necessity of the group aggregation mechanism. In WebShop, the average group size initially increases, suggesting that the agent's decisions converge during early training, leading to identical observations. Subsequently, the group size decreases because fine-grained credit assignment helps the agent identify and eliminate redundant steps; this leads to shorter interaction trajectories and fewer total observations. Conversely, in ALFWorld, the average group size increases after the reasoning format learning phase. While training still drives

Algorithm 1 Training LLM Agents with G2PO

Require: Initial policy $\pi_{\theta_{\text{old}}}$, reference policy π_{ref}, task distribution $p(X)$, discount factor γ, balancing weight w, clipping parameter ϵ, KL penalty β, sample size N, maximum steps T
for each training iteration **do**
  Update the old policy model: $\theta_{\text{old}} \leftarrow \theta$
  // Multi-step rollout phase
  Sample task $x \sim p(X)$ and initialize N identical environments
  for $j = 1$ to T **do**
    Sample actions $\left\{a_j^i \sim \pi_{\theta_{\text{old}}}(\cdot \mid o_j^i)\right\}_{i=1}^N$ (with $o_1^i = x$)
    Execute actions, observe next observations $\{o_{j+1}^i\}_{i=1}^N$
  end for
  Collect final outcome rewards $\{R_i\}_{i=1}^N$ for each trajectory τ_i
  // State group graph construction phase
  Partition intermediate observations $\mathcal{O}$ into G state groups $\{\mathcal{G}_k\}_{k=1}^G$
  Construct state transition graph $\{\mathcal{V}, \mathcal{E}\}$ with state groups as nodes and actions as edges
  // Group-aggregation state-value estimation phase
  Calculate discounted step values $v_j^i = \gamma^{T-j+1} R_i$ for all intermediate steps (Eq. 2)
  Estimate state group values $V(\mathcal{G}_k)$ by averaging step values within each group (Eq. 3)
  // Multi-granularity advantage estimation phase
  Compute group state-value differences δ_j^i (Eq. 5)
  Compute edge-centric advantages $A^{EC}(a_j^i)$ via group state-value differences δ_j^i (Eq. 6)
  Compute node-centric advantages $A^{NC}(a_j^i)$ via local reference sets Δ_k (Eq. 4)
  Compute episode-level advantages $A^{EP}(a_j^i)$ via outcome rewards R_i (Eq. 7)
  // Policy update phase
  Combine advantages: $A_j^i = A^{EP}(a_j^i) + w \cdot \left(A^{EC}(a_j^i) + A^{NC}(a_j^i)\right)$ (Eq. 8)
  Update policy θ by maximizing objective $\mathcal{J}(\theta)$ (Eq. 9)
end for

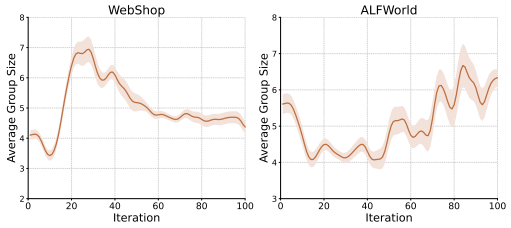


Figure 6: Dynamics of average group size during the training using Qwen2.5-1.5B-Instruct.

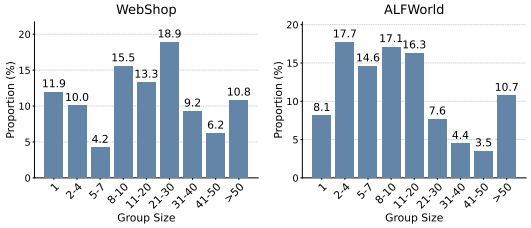


Figure 7: The distributions of group size using Qwen2.5-1.5B-Instruct

decision convergence, ALFWorld tasks inherently require repetitive actions, forcing the agent to revisit identical states. This inherent repetition amplifies the upward trend in group size, outweighing the downward trend caused by redundant step removal. Furthermore, as shown in Figure 7, it is evident that only 11.9% and 8.1% of steps in WebShop and ALFWorld, respectively, have a group size of 1. All remaining steps fall into groups of 2 or larger, meaning they all benefit from the reduced value estimation variance enabled by the group-aggregation mechanism.

For the more complex AppWorld benchmark, the mean state group size is 2.01 and the mean number of state groups per task graph is 155.38. Thus, overlap is lower than in WebShop and ALFWorld but remains non-trivial even in a long-horizon, API-based environment.

E Experiment Details

E.1 Descriptions of Benchmarks

We evaluate the performance of G2PO on three complex and realistic benchmarks. Here, we provide a brief introduction to these benchmarks.

WebShop. A comprehensive benchmark designed to evaluate the decision-making and language understanding capabilities of autonomous agents within a simulated e-commerce environment. It challenges agents to perform complex, multi-step tasks such as searching, navigating product pages, and selecting specific attributes (e.g., size, color, or style) to satisfy a given user requirement. Unlike static datasets, WebShop requires agents to interact with a dynamic interface, bridging the gap between linguistic instructions and functional web actions. It serves as a rigorous testing ground for assessing an agent's ability to reason and manage high-dimensional action spaces in realistic online scenarios.

ALFWorld. A large-scale benchmark for developing embodied AI agents that bridge the gap between language reasoning and physical execution. It maps high-level directives from the ALFRED dataset [42] into interactive, text-based environments using TextWorld. The benchmark features six categories of household tasks that require agents to perceive environment states, plan long-horizon trajectories, and generalize their knowledge across diverse indoor scenes.

AppWorld. AppWorld serves as a rigorous testing ground for LLM-based agents, providing a controllable yet realistic environment for tool-use and task planning. The benchmark comprises a rich suite of "apps" with a unified authentication system and a persistent database, simulating a multi-user world. It challenges agents with tasks that require deep API composition, long-horizon planning, and the ability to handle interleaved dependencies between different services. By utilizing an execution-based evaluation metric, where success is determined by the actual side effects on the environment rather than mere text similarity, AppWorld addresses the limitations of static evaluation and provides a more accurate assessment of an agent's reliability in real-world scenarios.

E.2 Comparing Baselines

We select nine baselines for comparison, including off-the-shelf models, prompt-based methods, and RL training approaches. Below, we provide a brief introduction to these baselines.

Qwen3.5-397B. A large-scale open-weights model with 397 billion parameters, known for its robust general capabilities and reasoning [54].

DeepSeek-V3.2. Another open-weights Mixture-of-Experts model optimized for efficiency, coding, and mathematical reasoning [24].

Gemini-2.5-Pro. A proprietary multimodal model featuring advanced long-context understanding and cross-modal reasoning [10].

ReAct. A framework that combines reasoning traces and task-specific actions to enable interleaved decision-making [56].

Reflexion. An agentic architecture that utilizes linguistic self-reflection and trial-and-error to iteratively improve task performance [41].

PPO. The standard policy gradient algorithm used for stable and efficient reinforcement learning from human feedback [39].

RLOO. An online RL method that leverages leave-one-out variance reduction to improve the efficiency of preference alignment [1].

GRPO A critic-free reinforcement learning approach that optimizes policies using relative rewards within sampled groups [40].

GiGPO. A reinforcement learning algorithm that achieves fine-grained multi-step credit assignment for LLM agents by estimating relative advantages at both the episode-level and the step-level via retroactively grouped identical environment states [13].

E.3 Training Details

Hyperparameters for ALFWorld. We apply consistent hyperparameters across all methods: the maximum lengths for prompts and responses are capped at 2048 and 512 tokens, respectively, with a maximum of 50 environment interaction steps per episode. The learning rates for the Actor and PPO critic are set to 1×10^{-6} and 1×10^{-5}. We employ a rule-based reward system where success receives a reward of 10, failure receives 0, and a penalty of -0.1 is applied to invalid actions. During the rollout

process, PPO utilizes 128 independent environments, whereas the group reinforcement learning methods perform sampling across 16 groups of 8 environments each (totaling 128 environments). The rollout and validation temperatures are set to 1.0 and 0.4, respectively. Other parameters include a mini-batch size of 256 and a KL divergence coefficient of 0.01. For G2PO, we set $\gamma = 0.95$ and fix $\omega = 1$.

Hyperparameters for WebShop. Identical hyperparameters are applied across all evaluated methods, with the maximum prompt and response lengths restricted to 4096 and 512 tokens, respectively. Each episode comprises a maximum of 15 environment steps. We set the learning rate at 1×10^{-6} for the actor and 1×10^{-5} for the critic (exclusive to PPO). A rule-based reward system is employed, where successes yield 10, failures yield 0, and invalid actions are penalized with -0.1. Following the ALFWorld setup, group-based RL methods utilize 128 environments in total (16 groups of 8 per rollout), whereas PPO employs 128 distinct environments. Temperatures for rollout and validation are maintained at 1.0 and 0.4. Furthermore, we use a mini-batch size of 64 and a KL-divergence loss coefficient of 0.01. For G2PO, the weighting coefficient ω is fixed at 1 without further tuning, alongside a discount factor γ of 0.95.

Hyperparameters for AppWorld. Identical hyperparameters are applied across all evaluated methods, with the maximum prompt and response lengths restricted to 13000 and 512 tokens, respectively. Each episode comprises a maximum of 30 environment steps. We set the learning rate at 1×10^{-6} for the actor and 1×10^{-5} for the critic (exclusive to PPO). A rule-based reward system is employed, where successes yield 10, failures yield 0, and invalid actions are penalized with -0.1. Following the ALFWorld setup, group-based RL methods utilize 128 environments in total (16 groups of 8 per rollout), whereas PPO employs 128 distinct environments. Temperatures for rollout and validation are maintained at 1.0 and 0.4. Furthermore, we use a mini-batch size of 64 and a KL-divergence loss coefficient of 0.01. For G2PO, the weighting coefficient ω is fixed at 1 without further tuning, alongside a discount factor γ of 0.95.

Computing Details. For ALFWorld and WebShop, We run Qwen2.5-1.5/7B experiments on 4×H100 GPUs, each for 100 iterations. For AppWorld, Qwen2.5-14B uses 8×H100 GPUs, for 50 iterations.

E.4 Prompts

Here we provide the prompts to guide reasoning and action of agents in RL on three benchmarks: WebShop (Figure 9), ALFWorld (Figure 8), and AppWorld (Figure 10).

Prompt Template for ALFWorld

You are an expert agent operating in the ALFRED embodied environment. Your task is to: {task_description}. Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}. You are now at step {current_step} and your current observation is: {current_observation}. Your admissible actions of the current situation are: [{admissible_actions}].
Now it's your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Figure 8: Prompt template for ALFWorld.

F Case Study

To demonstrate the superiority of edge-centric advantage (A^{EC}) over local group comparisons, we provide a case graph constructed during the advantage estimation phase of G2PO training on ALFWorld. As shown in Figure 11, to estimate the advantage of transitioning from state node 39 to node 7, the local group comparison method calculates the average value of all possible destination nodes as a baseline and measures the relative advantage of node 7's value against this average. While this provides a fine-grained credit assignment, it entirely ignores the intrinsic value of the source node and the actual value increment driven by the state transition. Consequently, this outcome-focused approach fails to evaluate the absolute contribution of an action. In contrast, A^{EC} evaluates an

Prompt Template for WebShop

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: {task_description}. Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}. You are now at step {current_step} and your current observation is: {current_observation}. Your admissible actions for the current situation are: [{available_actions}].
Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Figure 9: Prompt template for WebShop.

action's contribution to the overall goal by comparing the absolute value increments across all actions within the global graph, thereby overcoming the limited scope and destination-node dependency of local comparisons.

Specifically, the task is to *cool some bowl and put it in microwave*. The observation at node 39 is *You move the cup 1 to the microwave 1*, which completely deviates from the task objective. Conversely, the observation at node 7 is *You arrive at fridge 1. The fridge 1 is open. In it, you see nothing*, proving the agent has located the correct tool to *cool bowl*. This represents a massive leap forward toward the goal and thus merits a large positive advantage. However, under local group comparison, node 7 receives a marginal positive advantage of only 0.55 because its value is simply close to the average of all possible destination nodes. In contrast, A^{EC} accounts for the low value of the source node and standardizes it against the average value increments of all actions, assigning a substantial positive advantage of 2.1.

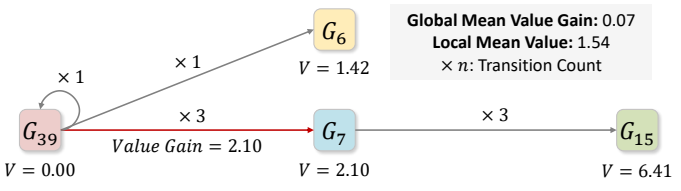
Prompt Template for AppWorld

I am your supervisor and you are a super intelligent AI Assistant whose job is to achieve my day-to-day tasks completely autonomously.
To do this, you will need to interact with app/s (e.g., spotify, venmo, etc) using their associated APIs on my behalf. For this you will undertake a *multi-step conversation* using a python REPL environment. That is, you will write the python code and the environment will execute it and show you the result, based on which, you will write python code for the next step and so on, until you've achieved the goal. This environment will let you interact with app/s using their associated APIs on my behalf.
Here are three key APIs that you need to know to get more information
# To get a list of apps that are available to you.
print(apis.api_docs.show_app_descriptions())
# To get the list of apis under any app listed above, e.g. supervisor
print(apis.api_docs.show_api_descriptions(app_name='supervisor'))
# To get the specification of a particular api, e.g. supervisor app's show_account_passwords
print(apis.api_docs.show_api_doc(app_name='supervisor',
api_name='show_account_passwords'))
The code will be executed in an interactive shell. Each code execution will produce an output that you can use in subsequent calls. Using these APIs, you can now generate code, that the environment will execute, to solve the task.

Here is an example:
...

Key Instructions and Disclaimers:
1. The email addresses, access tokens and variables (e.g. spotify_password) in the example above were only for demonstration. Obtain the correct information by calling relevant APIs yourself.
2. Only generate valid code blocks, i.e., do not put them in "'..."' or add any extra formatting. Any thoughts should be put as code comments.
3. You can use the variables from the previous code blocks in the subsequent code blocks.
...
16. Once you believe the task is complete, you MUST call 'apis.supervisor.complete_task()' to finalize it. If the task requires an answer, provide it using the answer argument — for example, 'apis.supervisor.complete_task(answer=<answer>)'. For tasks that do not require an answer, either omit the argument. The task will not end automatically — it will remain open until you explicitly make this call.
Using these APIs, now begin writing code cells step-by-step to solve the actual task:
My name is: {supervisor_first_name} {supervisor_last_name}. My personal email is {supervisor_email} and phone number is {supervisor_phone_number}.
Your task is: {task_description}
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} codes you generated and the corresponding environment return: {action_history}
Now you are at step {current_step} and it's your turn to generate code for this step. First, you MUST carefully reflect on the history of interactions and the most recent error messages. Then, reason about what should be done next, which APIs to call, what arguments to use, and how to build your code block to complete the task. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reflexion and reasoning, you present the solution code body within <code> </code> tags.

Figure 10: Prompt template for AppWorld.



Task: Cool some bowl and put it in microwave.

State	Observation
G_{39}	You move the cup 1 to the microwave 1
G_7	You arrive at fridge 1. The fridge 1 is open. In it, you see nothing.
G_{15}	You arrive at microwave 1. The microwave 1 is open. In it, you see a bowl 1.

Figure 11: A case study illustrating the superiority of the edge-centric advantage over local group comparisons. While local group comparisons provide the relative advantage between different actions within the same state, they fail to evaluate the absolute value improvement brought by an action from a global graph perspective.

G State Grouping under Observation Noise

Experimental setup. To test whether exact matching is essential, we construct a noisy ALFWorld variant by inserting task-irrelevant sentences, such as descriptions of outside noise or electromagnetic interference, into every returned observation. We train Qwen2.5-1.5B on this variant, merging observations when their Python `SequenceMatcher` similarity exceeds 0.95. We compare this setting with exact matching on the original observations without noise.

Results. Table 3 reports the success rates for both settings. Similarity matching under observation noise achieves an overall success rate of 95.3%, compared with 95.0% for exact matching without noise. Although individual task categories vary, overall performance is preserved. This result supports the flexibility of G2PO beyond byte-identical observations when nuisance text is present.

Table 3: Success rates (%) on ALFWorld under different observation and grouping settings.

Setting	Pick	Look	Clean	Heat	Cool	Pick2	All
No noise + exact matching	97.2	96.3	95.1	94.9	97.1	91.4	95.0
Noise + similarity matching	94.0	81.8	100.0	100.0	100.0	88.9	95.3

H Empirical Comparison with GEPO

GEPO [59] uses $N = 16$ rollouts per group in its main experiments, whereas all main-paper G2PO results use $N = 8$. We therefore additionally train G2PO with $n = 16$ on ALFWorld using Qwen2.5-1.5B, and compare the $n = 8$ setting on WebShop using Qwen2.5-7B, for which GEPO also reports results. The GEPO results below are taken from its reported experiments.

Table 4: Success rates (%) on ALFWorld, comparing GEPO and G2PO.

Method	Pick	Look	Clean	Heat	Cool	Pick2	All
GEPO ($N = 16$)	99.2	84.5	96.8	88.5	83.6	92.8	92.9
G2PO ($N = 8$)	97.2	96.3	95.1	94.9	97.1	91.4	95.0
G2PO ($N = 16$)	100.0	100.0	100.0	100.0	89.5	95.2	97.7

As shown in Table 4, G2PO exceeds GEPO on ALFWorld by 2.1 percentage points while using half as many rollouts per group (95.0% vs. 92.9%). At the matched rollout count of $N = 16$, G2PO achieves 97.7%, a gain of 4.8 points over GEPO.

Table 5: Success rates (%) on WebShop with Qwen2.5-7B.

Method	Success rate
GEPO ($N = 8$)	78.5
G2PO ($N = 8$)	78.3

On WebShop, the two methods achieve comparable success rates at $N = 8$, differing by only 0.2 percentage points (Table 5). GEPO reports additional training overhead of approximately 15% at $N = 8$ and 35% at $N = 16$, while the overhead measured for G2PO in our main-paper setting is approximately 0.4% per training step (Figure 5). These results show comparable or higher success rates for G2PO across the evaluated settings, with substantially lower reported computational overhead.

I Diagnostic Validation of the Edge-Centric Advantage

We use ALFWorld and Qwen2.5-1.5B to examine the edge-centric advantage from two perspectives: the credit assigned to actions on a shortest path to success, and its correlation with action importance.

I.1 Advantages of Critical and Non-Critical Actions

For each constructed task graph, we find a shortest path from the initial state to a success state, label actions on this path as critical, and label the remaining actions as non-critical. For this diagnostic, we report the unnormalized edge-centric advantage, denoted A^{EC}_{raw}, before the global standardization in Eq. 6. We compare its average over critical and non-critical actions within each task.

Table 6: Unnormalized edge-centric advantages for critical and non-critical actions on ALFWorld.

Metric	Value
Tasks where mean critical A^{EC}_{raw} exceeds mean non-critical A^{EC}_{raw}	91.0%
Average critical A^{EC}_{raw}	$+0.7049$
Average non-critical A^{EC}_{raw}	$+0.2394$
Average difference (critical $-$ non-critical)	$+0.4655$

As shown in Table 6, critical actions receive higher average advantages in 91.0% of tasks. Their average advantage is nearly three times that of non-critical actions, with a mean difference of $+0.4655$. This diagnostic supports the ability of edge-centric credit assignment to prioritize actions on short paths to task completion.

I.2 Correlation with Action Importance

We define an action's importance as the task success rate following its execution. We calculate the episode-level, node-centric, and edge-centric advantages and measure their Pearson correlation r and Spearman rank correlation ρ with action importance.

Table 7: Correlation between advantage estimates and action importance on ALFWorld.

Advantage method	Pearson r	Spearman ρ
Episode-level	$+0.0304$	$+0.0238$
Node-centric	$+0.0492$	$+0.0625$
Edge-centric	$+0.2531$	$+0.1636$

Table 7 shows that the edge-centric signal has approximately $5.1\times$ and $8.3\times$ the Pearson correlation of the node-centric and episode-level signals, respectively, and $2.6\times$ and $6.9\times$ their Spearman correlation. Although the absolute correlations are modest, the stronger association supports the ability of the edge-centric signal to identify actions associated with higher task success rates.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As stated in abstract and introduction, our contributions and scope can be accurately reflected by main claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are discussed in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the formal proof for the variance reduction of our proposed method in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full description of our method in Section 4 and experimental setting in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The dataset is from existing publicly available benchmark and we will release code at publication time.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide relevant details in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our results in Table 1 are accompanied by standard errors.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide computer resources in Section E.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We do not foresee that our research has any potential major ethical problem.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work is a foundational research, which would bring contributions to research community. We believe the proposed method has no direct societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper has no such risks as it didn't release generative models or scraped datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the assets used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We only use LLMs for polishing the writing, e.g., checking for grammatical mistakes or typos.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.