STRUCTURE-GUIDED POLICY OPTIMIZATION FOR LONG-HORIZON LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown promise as sequential decision-makers, yet they still struggle in long-horizon tasks where sparse terminal rewards provide limited guidance for intermediate actions. Existing optimization methods mainly densify supervision through trajectory-level aggregation or local scalar proxies, often treating multi-turn interactions as flat sequences. As a result, they lack an explicit coordinate system for measuring task progress, making it difficult to distinguish redundant local exploration from structural transitions that genuinely advance the task. We propose Structure-Guided Policy Optimization (SGPO), a framework that grounds policy learning in the empirical topology of agent-environment interactions. SGPO aggregates self-generated trajectories into a frequency-weighted state transition graph and partitions it into topological regions that capture latent task structure. Anchored by successful outcomes, this recovered structure induces hierarchical guidance signals: an inter-regional signal that encourages progression across structural bottlenecks toward the target region, and an intra-regional signal that guides efficient local navigation toward preferred exits. These normalized structural signals are integrated at the advantage level with group-based outcome supervision, providing dense stepwise feedback while preserving task success as the primary optimization objective. Extensive experiments on ALFWorld, WebShop and Sokoban show that SGPO improves performance and training stability across diverse LLM scales. The source code is available in the supplementary.

1 INTRODUCTION

Large language models are increasingly deployed as agents that interact with environments over extended sequences of perception, reasoning and action (Yao et al., 2022b; Guo et al., 2025). Beyond single-turn generation, these agents now operate in multi-turn settings such as embodied navigation (Shridhar et al., 2020; Wu et al., 2024), interactive games (Zhu et al., 2023; Hu et al., 2024) and web-based task execution (Gur et al., 2023; He et al., 2024), where success emerges only after a long sequence of decisions. In such long-horizon environments, the agent must continuously evaluate whether its actions are meaningfully advancing the task, despite receiving sparse and delayed reward signals that provide little immediate guidance (Wang et al., 2023; Zhang et al., 2025). This introduces a fundamental challenge distinct from short-horizon decision making: progress toward the goal is implicit, delayed and difficult to evaluate from local observations alone. This renders multi-turn interaction not merely a longer version of single-step reasoning, but a qualitatively different regime in which the agent must act without a clear reference to its global standing in the task process.

To make multi-turn decision-making easier, previous studies have focused on providing agents with clearer feedback. These methods typically address the problem of delayed rewards by adding intermediate incentives (Ng et al., 1999; Devlin & Kudenko, 2012), evaluating each step based on the final outcome of the entire sequence (Andrychowicz et al., 2017), or comparing multiple attempts to identify which paths are more effective (Shao et al., 2024). Such strategies have shown promise in stabilizing optimization and reducing variance in long-horizon settings. However, despite these advances, a critical limitation remains overlooked. Many existing methods treat multi-turn interactions as structurally homogeneous sequences, assigning intermediate learning signals via trajectory-level aggregation or scalar proxies. As a result, these signals lack an explicit coordinate of progress that reflects how individual actions contribute to advancement toward task completion, making it difficult for the agent to distinguish between actions that merely induce trivial local variations and those

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

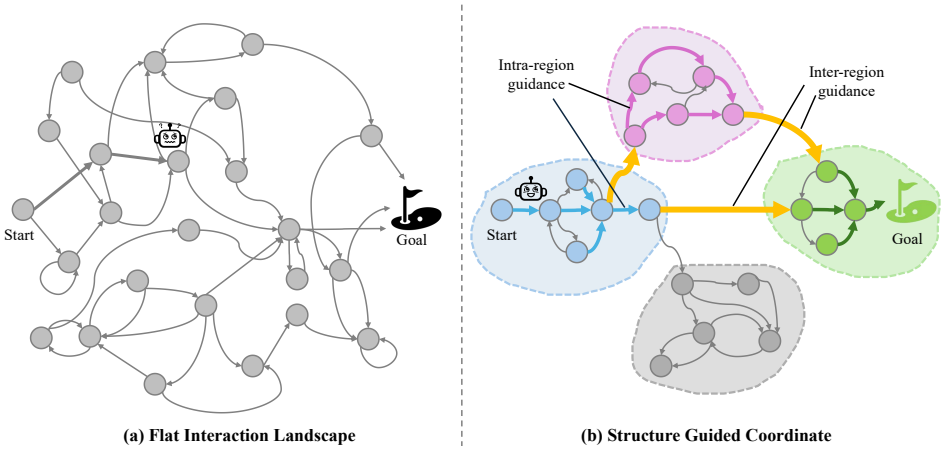


Figure 1: From flat sequences to structure-guided coordinate. (a) The raw interaction landscape presents a confusing, unstructured graph where progress is implicit. (b) SGPO extracts a macroscopic skeleton from experience, revealing structural regions and bottlenecks. This topological awareness enables hierarchical guidance, explicitly differentiating between intra-regional navigation and inter-regional progression.

that drive meaningful structural transitions. This leads to inefficient exploration and brittle credit assignment, especially when success implicitly requires navigating through multiple latent stages with distinct roles in the overall task-solving process.

Multi-turn tasks are inherently structured, yet this structure is typically absent from the learning process. In many long-horizon environments, task execution naturally unfolds through a sequence of latent stages: within a stage, states are closely related, and actions induce local variations, whereas transitions across stages represent bottleneck transitions that advance the task. However, many existing formulations overlook this hierarchy, largely treating transitions as equivalent steps in a flat sequence. This conceptual flattening obscures the critical distinction between intra-stage navigation and inter-stage progression, leaving task progress implicit. To overcome this ambiguity, effective multi-turn learning requires making the task structure explicit, enabling progress to be defined relative to the agent's position within it rather than inferred solely from delayed outcomes.

To address this challenge, we propose **Structure-Guided Policy Optimization (SGPO)**, a framework that leverages self-generated experience to ground policy learning in the task's empirical topology without relying on external supervision or predefined structural annotations. The central idea is to recover latent task structure by aggregating interaction trajectories into a state transition graph and partitioning states into structural regions that capture distinct task stages. Unlike methods that process interactions as flat sequences, SGPO utilizes this graph as a structured coordinate system for characterizing the relative progress of individual decisions. Based on this representation, we derive hierarchical progress signals that operate at two scales: guiding coarse-grained inter-regional progression and optimizing fine-grained intra-regional navigation. By aligning policy optimization with the recovered task structure, SGPO provides a structure-aware mechanism to stabilize credit assignment and encourage structure-consistent progress in long-horizon tasks.

In summary, our contributions are as follows:

- We identify the absence of explicit task structure as a key bottleneck in long-horizon decision making. We argue that long-horizon difficulty reflects the lack of a progress coordinate system rather than solely reward sparsity.
- We propose SGPO, a framework that grounds policy optimization in recovered empirical task structure. By aggregating self-generated experience into a transition graph, SGPO derives inter- and intra-regional progress signals to provide dense and structure-consistent guidance.
- We empirically demonstrate that modeling task structure improves long-horizon learning. Across multiple benchmarks, our method outperforms strong baselines in success rate and final performance, while exhibiting more stable training behavior.

2 Related Work

2.1 Credit Assignment in Agentic RL

Credit assignment remains a fundamental challenge in agentic reinforcement learning, particularly in multi-turn settings with sparse rewards (Sutton, 1988). To mitigate this, existing approaches primarily focus on densifying learning signals by aggregating supervision over extended interactions. Representative strategies include leveraging process-level reward models (Chae et al., 2025; Xi et al., 2025; Wang et al., 2025b) or preference signals to provide intermediate guidance (Cui et al., 2025; Liu et al., 2025), shaping exploration through information-theoretic objectives such as information gain (Wang et al., 2025a), and encouraging behavioral diversity via entropy regularization (Dong et al., 2025; Lin et al., 2025). In addition, methods like GiGPO (Feng et al., 2025) introduce anchor states to stabilize relative evaluation across trajectories. Despite their methodological differences, these approaches share a common limitation: task progress is approximated through scalar proxies or local heuristics that function independently of the task's underlying structure. Under this paradigm, credit assignment is driven by preference consistency or exploration incentives, rather than by an explicit representation of the task's structural transitions. This gap limits the ability of learning signals to differentiate between structural transitions and local variations, leading to inefficient optimization in complex long-horizon environments.

2.2 Task Structure in Sequential Decision Making

A related line of work studies task structure through hierarchical or stage-based views of sequential decision making, where complex tasks are decomposed into multiple stages or sub-policies to facilitate planning. Hierarchical reinforcement learning (Kulkarni et al., 2016) and option-based frameworks (Bacon et al., 2017) explicitly model such structure to enable temporal abstraction and reusable behaviors. While these approaches assume that tasks unfold through latent stages, the resulting structure is primarily leveraged for hierarchical control and action selection, rather than for defining task progress or guiding credit assignment in multi-turn learning. Another closely related direction focuses on discovering subgoals from interaction trajectories or demonstrations (McGovern & Barto, 2001; Chen et al., 2021). While these methods effectively reveal latent task structure, the resulting insights are typically used to construct discrete intermediate objectives or behavioral primitives, and are rarely integrated into the dense valuation of long-horizon trajectories. In contrast to these approaches, our work treats the recovered region-level task structure as an explicit coordinate system for evaluating progress and assigning credit across extended interactions.

3 Preliminaries

3.1 Problem Formulation

We formalize long-horizon decision making as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$. At each time step t, the agent is in a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$ according to a policy $\pi_\theta(a_t \mid s_t)$, and the environment transitions to $s_{t+1} \sim \mathcal{T}(\cdot \mid s_t, a_t)$ while emitting a scalar reward r_t. An episode induces a trajectory $\tau = (s_0, a_0, r_0, \ldots, s_T, a_T, r_T)$, and the objective is to maximize the expected discounted return $J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$. A primary challenge in this setting is the sparsity of rewards. In many tasks, informative signals are only provided upon goal completion, leading to a difficult credit assignment problem. Without dense intermediate feedback, it is challenging for the agent to identify which specific actions in a long trajectory contribute to the final outcome.

3.2 Group Relative Policy Optimization (GRPO)

We utilize GRPO (Shao et al., 2024) as our optimization backbone. GRPO eliminates the value critic by estimating baselines from a group of trajectories $\{\tau_i\}_{i=1}^{G}$ sampled for the same input. The advantage for each trajectory is derived by normalizing its return R_i against the group statistics:

$$A^O(\tau_i) = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G})}. \tag{1}$$

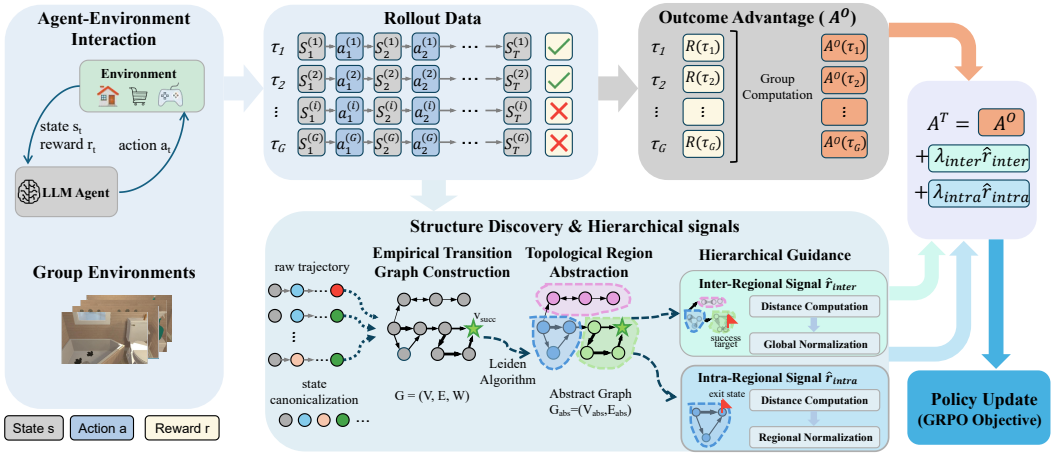


Figure 2: **Overview of SGPO.** SGPO augments standard group-based policy optimization with structure-aware guidance. A batch of interaction trajectories is collected by the agent, from which a sparse outcome advantage (A^O) is computed. In parallel, the same experience is aggregated into an empirical transition graph to recover latent topological regions. This recovered structure serves as a coordinate system to derive dense hierarchical signals: inter-regional guidance ($\hat{r}_{\text{inter}}$) for traversing structural bottlenecks and intra-regional guidance ($\hat{r}_{\text{intra}}$) for efficient local navigation.

The policy is then updated via a PPO-style surrogate objective using these relative advantages (Schulman et al., 2017). While effective for stability, this formulation relies heavily on the scalar returns R to provide a sufficient discriminative signal for distinguishing trajectories within a group.

4 Method

4.1 Constructing the Empirical Transition Graph

In multi-turn decision-making, individual trajectories inherently provide only a local, linear view of the task. Each trajectory reflects a single sequence of decisions taken under a particular policy (Li et al., 2025; Yuan et al., 2025), revealing limited information about alternative pathways or environmental connectivity, thereby obscuring the higher-level patterns that underlie long-horizon progress.

To expose these latent patterns, we aggregate experience from dispersed trajectories into a unified structural representation. Across different rollouts, agents frequently revisit states and traverse overlapping transitions. These overlaps allow connectivity patterns—such as recurring states, branching points and bottlenecks—to become explicit, revealing global organization invisible to single executions. We formalize this aggregated structure as the **Empirical Transition Graph**, a discrete approximation of the empirical task topology induced by interaction data.

From Raw Trajectories to Canonical States. Let $\mathcal{D} = \{\tau_i\}_{i=1}^{G}$ be a batch of group-sampled trajectories. In text-based environments, raw observations are often partially observable. To address this, we represent the effective state s_t as a context-augmented observation (detailed construction and discussion provided in Appendix A.1).

We project environment states onto a set of canonical nodes $\mathcal{V}$ using a state abstraction mapping ϕ. In practice, ϕ can be instantiated by different matching rules, such as exact matching over context-augmented observations or similarity-based consolidation, to capture recurrent or semantically related interaction states. Formally, the node set is defined as:

$$\mathcal{V} = \bigcup_{\tau \in \mathcal{D}} \bigcup_{t=0}^{|\tau|} \{\phi(s_t)\}. \tag{2}$$

This formulation explicitly captures state re-visitation. To ensure topological consistency across diverse successful strategies, we append a *virtual success node* v_{succ} to the end of every successful

trajectory. This augmentation unifies the task termination condition, ensuring that all empirically successful paths converge to a single sink state regardless of their intermediate behavioral variations.

Establishing Frequency-Weighted Edges. Having mapped raw observations to canonical nodes $\mathcal{V}$, we transform linear trajectories into a directed transition network. For every trajectory $\tau \in \mathcal{D}$, we extract all adjacent effective-state pairs (s_t, s_{t+1}) to identify directed transitions between canonical nodes. We define the edge set $\mathcal{E}$ as all transitions observed at least once:

$$\mathcal{E} = \{(u, v) \mid \exists \tau \in \mathcal{D},\, t : \phi(s_t) = u,\ \phi(s_{t+1}) = v\}. \tag{3}$$

Our construction is frequency-aware: we assign each edge (u, v) a weight w_{uv} equal to its empirical count across trajectories,

$$w_{uv} = \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{|\tau|-1} \mathbb{I}[\phi(s_t) = u \wedge \phi(s_{t+1}) = v]\,, \tag{4}$$

where $\mathbb{I}[\cdot]$ is the indicator function. This yields the empirical transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with weights $\mathcal{W} = \{w_{uv} : (u, v) \in \mathcal{E}\}$. Here, w_{uv} provides an objective measure of empirical support: it records how often a transition is observed, without imposing assumptions about optimality.

The resulting graph $\mathcal{G}$ acts as a data-driven skeleton of the environment, capturing empirical reachability rather than optimality. This neutral topology lays the foundation for deriving structure-aware metrics to evaluate progress in subsequent steps.

4.2 Topological Region Abstraction

While the empirical transition graph $\mathcal{G}$ reveals global connectivity, it remains inherently fine-grained. In complex text-based environments, the state space is populated with redundant variations: numerous canonical nodes differ by local details yet correspond to the same latent structural region. Reasoning directly at this microscopic level obscures the macroscopic organization of the environment. To distill the latent task structure reflected by empirical connectivity, we advance to a region-level abstraction. Our objective is to partition the graph into coherent structural regions based on topological connectivity, ensuring that the extracted structure captures the task's intrinsic dynamics rather than mere surface-level semantic similarity.

A **Structural Region** $\mathcal{C} \subset \mathcal{V}$ is a cohesive subgraph characterized by high intra-regional transition density and homogeneous connectivity patterns. Conversely, transitions bridging distinct regions constitute sparse topological bottlenecks within the global network $\mathcal{G}$. Formally, we seek a partition of the node set $\mathcal{V}$ into K disjoint regions, denoted as $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, such that:

$$\mathcal{V} = \bigcup_{k=1}^{K} \mathcal{C}_k, \quad \text{and} \quad \mathcal{C}_i \cap \mathcal{C}_j = \emptyset,\ \forall i \neq j. \tag{5}$$

We instantiate region discovery via a modularity-based criterion (Newman & Girvan, 2004; Newman, 2006): nodes within each region $\mathcal{C}_k$ should exhibit strong mutual connectivity relative to a random null model. This is captured by a global quality function $Q(\mathcal{P})$ of the form

$$Q(\mathcal{P}) = \sum_{\mathcal{C}_k \in \mathcal{P}} \left(\text{mass}(\mathcal{C}_k) - \mathbb{E}[\text{mass}(\mathcal{C}_k)]\right), \tag{6}$$

where $\text{mass}(\mathcal{C}_k)$ is the total transition weight within region $\mathcal{C}_k$, and $\mathbb{E}[\cdot]$ is the expected mass under a null model of random connectivity. Under this abstraction, each region functions as a macroscopic node, grouping local variations while separating regions connected by structural transitions.

Direct optimization of Q is intractable for large graphs. We therefore adopt the Leiden algorithm (Traag et al., 2019) to efficiently approximate a high-quality partition. Leiden guarantees well-connected communities and converges efficiently in practice, enabling systematic extraction of $\mathcal{P}$ from interaction data.

Upon obtaining the partition $\mathcal{P}$, we construct the **Abstract Task Graph** $\mathcal{G}_{abs} = (\mathcal{V}_{abs}, \mathcal{E}_{abs})$, with $\mathcal{V}_{abs} = \mathcal{P}$. A directed edge $(\mathcal{C}_i, \mathcal{C}_j) \in \mathcal{E}_{abs}$ exists if there is at least one observed transition $u \to v$ in $\mathcal{G}$ such that $u \in \mathcal{C}_i$ and $v \in \mathcal{C}_j$. $\mathcal{G}_{abs}$ provides a coarse region-level view of connectivity that we use to define inter-regional progression and intra-regional navigation signals.

4.3 HIERARCHICAL STRUCTURAL GUIDANCE

The regional abstraction $\mathcal{G}_{abs}$ derived in the previous section reveals that agent-environment interactions are not uniform, but rather exhibit two qualitatively distinct structural regimes: *intra-regional variations* (dense, high-frequency transitions representing local state perturbations) and *inter-regional transitions* (sparse, directional jumps that traverse topological bottlenecks). Treating these heterogeneous dynamics as equivalent steps in a flat credit assignment scheme obscures the distinction between locally reversible behavior and genuine structural progress, making it difficult for the agent to reason about advancement in long-horizon tasks.

To address this mismatch, we propose a hierarchical framework that constructs distinct guidance signals for macroscopic flow and microscopic navigation. The recovered topological structure is obtained unsupervised and is therefore semantically neutral: while it captures empirical connectivity, it does not yet encode which directions correspond to meaningful task advancement. To ground this neutral structure in task semantics, we leverage the virtual success node v_{succ} to identify success-reachable regions and induce an orientation over the abstract graph. This sink-anchored orientation establishes a notion of structural directionality, forming the basis for the hierarchical guidance signals.

Inter-Regional Progression. To guide the agent through sparse topological bottlenecks, we first construct a directed skeleton over the neutral regions. We identify the target region $\mathcal{C}_{\text{target}} \in \mathcal{P}$ as the unique region containing the virtual success node v_{succ}. Using this target as the global anchor, we compute the topological distance $d_{inter}(\mathcal{C})$ for each region $\mathcal{C} \in \mathcal{P}$, defined as the shortest path distance to $\mathcal{C}_{\text{target}}$ on the abstract graph $\mathcal{G}_{abs}$.

This process serves a dual purpose: orientation and filtering. First, $d_{inter}(\mathcal{C})$ establishes a relative structural organization by quantifying the topological bottlenecks separating a given region from the success target. This distance is computed via a reverse Breadth-First Search (BFS) starting from $\mathcal{C}_{\text{target}}$, effectively isolating the success-reachable skeleton. Regions inaccessible from the target in the reverse view are masked out from the structural coordinates, and transitions entering these unsupported areas receive a conservative negative signal. This mechanism discourages the agent from drifting into regions not validated by successful experience.

Based on this distance-based organization, we derive the inter-regional guidance signal. For a transition by action a_t between distinct regions $s_t \in \mathcal{C}_i \to s_{t+1} \in \mathcal{C}_j$ (where $i \neq j$), the agent receives a structural flow reward defined as the reduction in topological distance:

$$r_{inter}(a_t) = d_{inter}(\mathcal{C}_i) - d_{inter}(\mathcal{C}_j) \tag{7}$$

By explicitly anchoring the coordinate system to the virtual success sink, r_{inter} provides a consistent global directional signal that steers the agent towards task completion, bridging the temporal gap between sparse environmental rewards.

Intra-Regional Navigation. Within a structural region $\mathcal{C}$, the macroscopic topological distance $d_{inter}(\mathcal{C})$ remains constant, rendering the inter-regional flow signal uninformative for local action selection. This stationarity often leaves the agent prone to aimless cycling amidst dense local variations. To address this, we construct a microscopic guidance signal that functions as a local compass, steering the agent towards the boundaries that facilitate structural advancement.

We first identify the set of preferred exits $\mathcal{E}_{pref}(\mathcal{C})$ for the current region. An exit state is considered preferred if it bridges to a neighboring region $\mathcal{C}'$ with a strictly lower topological distance. This criterion ensures that the local navigation is consistently aligned with the macroscopic flow derived in the previous step. Based on these exits, we compute a local distance metric. Specifically, we perform a reverse BFS within the region's internal transition graph, starting from the preferred exits. For each state $s \in \mathcal{C}$, we calculate the local distance $d_{intra}(s)$, defined as the minimum number of steps required to reach any state in $\mathcal{E}_{pref}(\mathcal{C})$. For a transition by action a_t from $s_t \to s_{t+1}$ within the same region, the intra-regional guidance reward is formulated as the reduction in this local distance:

$$r_{intra}(a_t) = d_{intra}(s_t) - d_{intra}(s_{t+1}) \tag{8}$$

By incentivizing reduction of d_{intra}, this signal encourages efficient navigation toward exits structurally aligned with inter-regional progression, reducing redundant interactions within the region.

Scale-Adaptive Integration for Policy Optimization. The macroscopic inter-regional signals and microscopic intra-regional navigation signals arise from structurally distinct types of transitions and

therefore exhibit different statistical characteristics. Treating these signals as directly comparable and summing them naively can lead to unstable optimization, as progress across regions and local navigation within regions contribute unevenly to the learning signal. To ensure consistent and balanced credit assignment, we normalize these signals adaptively based on their specific structural contexts before integration.

To address this, we implement a region-conditioned normalization strategy. For the intra-regional signal r_{intra} computed over the action a_t, we compute region-specific statistics (mean $\mu_{\mathcal{C}}$ and standard deviation $\sigma_{\mathcal{C}}$) conditioned on the current region $\mathcal{C}(s_t)$. The inter-regional signal r_{inter} is normalized using global statistics ($\mu_{glob}, \sigma_{glob}$). The normalized structural signals are computed as:

$$\hat{r}_{intra}(a_t) = \frac{r_{intra}(a_t) - \mu_{\mathcal{C}(s_t)}}{\sigma_{\mathcal{C}(s_t)}}, \tag{9}$$

$$\hat{r}_{inter}(a_t) = \frac{r_{inter}(a_t) - \mu_{glob}}{\sigma_{glob}}. \tag{10}$$

This converts absolute distance reductions into relative structural scores, yielding more stable and comparable learning signals across different structural contexts. In practice, a small stability constant ϵ is added to the denominator to prevent division by zero.

Finally, we integrate these dense structural signals into the GRPO framework. These signals do not modify the environment reward directly; instead, they serve as outcome-anchored auxiliary corrections to the advantage, providing structure-aware step-level guidance on top of group-based outcome supervision. Following GRPO, the trajectory-level outcome advantage $A^O(\tau_i)$ in Eq. (1) is shared across all actions within the trajectory, while structural scores provide step-specific refinements. The total advantage used for policy optimization is:

$$A^T(a_t, \tau_i) = A^O(\tau_i) + \lambda_{\text{intra}}\hat{r}_{\text{intra}}(a_t) + \lambda_{\text{inter}}\hat{r}_{\text{inter}}(a_t), \tag{11}$$

where λ_{intra} and λ_{inter} are coefficients balancing the influence of microscopic navigation and macroscopic progression. This formulation can be viewed as a structure-guided surrogate advantage: the outcome advantage anchors learning to task success, while the structural terms redistribute step-level credit toward transitions aligned with the recovered task topology.

5 Experiments

5.1 Experiment Setup

Benchmarks. We conduct evaluations on three distinct multi-turn benchmarks covering both text-based and image-based domains. **ALFWorld** (Shridhar et al., 2020) is an embodied environment that simulates 3D household tasks via text interactions to assess long-horizon planning, covering six specific task categories: Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place and Pick2 & Place. **WebShop** (Yao et al., 2022a) is a large-scale e-commerce environment featuring a vast state space with millions of product listings, where agents must navigate web pages through searching and clicking to purchase specific items. Additionally, we evaluate on **Sokoban** (Schrader, 2018), a classic image-based grid puzzle in which agents must analyze visual layouts and push boxes to target locations, requiring spatial reasoning and forward planning to handle obstacles and irreversible states.

Baselines. For ALFWorld and WebShop, we compare SGPO against a diverse set of competitive baselines, following the evaluation protocol of GiGPO (Feng et al., 2025). Specifically, we consider: (1) **Closed-source LLMs**: GPT-4o (Achiam et al., 2023) and Gemini-2.5-Pro (Team et al., 2023), evaluated as strong off-the-shelf language agents. (2) **Prompting-based agents**: ReAct (Yao et al., 2022b) and Reflexion (Shinn et al., 2023), which guide multi-step interaction via in-context prompting without parameter updates. (3) **RL-trained methods**: PPO (Schulman et al., 2017), a standard actor–critic algorithm with a learned value function, and critic-free methods including RLOO (Kool et al., 2019; Ahmadian et al., 2024), GRPO (Shao et al., 2024) and GiGPO. For Sokoban, we compare SGPO with group-based methods GRPO and GiGPO. To ensure a fair comparison, results for all baselines follow the evaluation protocol of GiGPO, maintaining identical experimental settings including base models and training budgets.

Table 1: Performance on ALFWorld and WebShop. For ALFWorld, we report the average success rate (%) for each subtask as well as the overall result. For WebShop, we report both the average score and the average success rate (%).

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training	PPO (with critic)	$64.8_{\pm 3.5}$	$40.5_{\pm 6.9}$	$57.1_{\pm 4.9}$	$60.6_{\pm 6.6}$	$46.4_{\pm 4.0}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$	$73.8_{\pm 3.0}$	$51.5_{\pm 2.9}$
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	GRPO	$85.3_{\pm 1.5}$	$53.7_{\pm 8.0}$	$84.5_{\pm 6.8}$	$78.2_{\pm 7.9}$	$59.7_{\pm 5.0}$	$53.5_{\pm 5.6}$	$72.8_{\pm 3.6}$	$75.8_{\pm 3.5}$	$56.8_{\pm 3.8}$
RL Training	GiGPO	$96.0_{\pm 1.4}$	$76.5_{\pm 3.9}$	$91.8_{\pm 5.5}$	$91.3_{\pm 6.3}$	$71.7_{\pm 8.4}$	$\mathbf{79.5}_{\pm 7.7}$	$86.1_{\pm 4.7}$	$83.5_{\pm 1.8}$	$67.4_{\pm 4.5}$
RL Training	SGPO	$\mathbf{97.9}_{\pm 3.6}$	$\mathbf{86.5}_{\pm 6.9}$	$\mathbf{100.0}_{\pm \mathbf{0.0}}$	$\mathbf{94.9}_{\pm 8.9}$	$\mathbf{77.7}_{\pm 12.0}$	$74.4_{\pm 9.1}$	$\mathbf{89.6}_{\pm 4.3}$	$\mathbf{85.6}_{\pm 2.0}$	$\mathbf{71.6}_{\pm 2.9}$
Qwen2.5-7B-Instruct										
Prompting	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
RL Training	PPO (with critic)	$92.3_{\pm 4.0}$	$64.0_{\pm 8.4}$	$92.5_{\pm 2.4}$	$89.5_{\pm 7.0}$	$80.3_{\pm 2.0}$	$68.8_{\pm 8.3}$	$80.4_{\pm 2.7}$	$81.4_{\pm 3.1}$	$68.7_{\pm 5.1}$
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	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	GiGPO	$91.8_{\pm 5.4}$	$\mathbf{88.6}_{\pm 6.3}$	$95.9_{\pm 3.2}$	$90.2_{\pm 2.6}$	$86.5_{\pm 5.5}$	$\mathbf{85.2}_{\pm 7.5}$	$90.2_{\pm 2.3}$	$86.2_{\pm 2.6}$	$75.2_{\pm 3.8}$
RL Training	SGPO	$\mathbf{100.0}_{\pm \mathbf{0.0}}$	$87.0_{\pm 17.9}$	$\mathbf{96.0}_{\pm 3.9}$	$\mathbf{98.1}_{\pm 3.2}$	$\mathbf{91.9}_{\pm 2.2}$	$82.2_{\pm 5.5}$	$\mathbf{93.8}_{\pm 2.7}$	$\mathbf{87.5}_{\pm 1.4}$	$\mathbf{77.2}_{\pm 3.4}$

Training Details. We implement SGPO using the VERL framework (Feng et al., 2025). For the backbone models, we employ Qwen2.5-1.5B/7B-Instruct (Bai et al., 2025) for text-based benchmarks (ALFWorld and Webshop), and Qwen2.5-VL-3B-Instruct for the vision-based Sokoban task to process grid-layout observations. To ensure a fair comparison with the baselines, we maintain a consistent group size of $G = 8$ and identical interaction protocols. The structural guidance coefficients are empirically set to $\lambda_{inter} = 1$ and $\lambda_{intra} = 1$ to balance macroscopic progression and microscopic navigation. For more comprehensive details on computation overhead, hyperparameter configurations and prompt templates, please refer to Appendix B.

5.2 Overall Performance

Table 1 reports the performance of SGPO in comparison with prompting-based methods, standard RL approaches, and group-based RL baselines on ALFWorld and WebShop, while Table 2 presents results on the image-based Sokoban task. All results are averaged over three random seeds. Across all benchmarks and model scales, SGPO achieves the highest overall performance among the evaluated methods. In particular, SGPO outperforms existing group-based RL approaches, such as GRPO and GiGPO across most evaluation metrics, indicating that incorporating recovered task structure provides additional benefits beyond trajectory-level grouping alone.

Table 2: Performance comparison on Sokoban [6×6].

Type	Method	Sokoban [6×6]
Prompting	Qwen2.5-VL	11.7
RL Training	GRPO	$67.1_{\pm 4.7}$
RL Training	$\text{GiGPO}_{\text{w/ std}}$	$76.9_{\pm 2.7}$
RL Training	$\text{GiGPO}_{\text{w/o std}}$	$81.0_{\pm 3.6}$
RL Training	**SGPO**	$\mathbf{82.1}_{\pm 2.8}$

On ALFWorld, SGPO attains the highest overall success rate across both 1.5B and 7B scales, achieving 100% success on the *Clean* task (1.5B) and the *Pick* task (7B). These results reflect stable performance on tasks that require consistent multi-step execution. For the more challenging *Cool* and *Heat* categories, SGPO improves over the strongest baseline, GiGPO, by a margin of approximately 5%–10%. On the *Pick2* subtask, SGPO performs competitively but slightly underperforms GiGPO. This marginal gap is likely attributable to subtle differences in context-augmented state representation and canonicalization. We discuss these differences in Appendix A.1.

On WebShop, SGPO achieves the best performance across both model scales, with the highest average score and success rate. With the 1.5B model, SGPO improves the average score from 83.5 to 85.6 and the success rate from 67.4% to 71.6% compared to GiGPO. With the 7B model, SGPO further increases the average score from 86.2 to 87.5 and the success rate from 75.2% to 77.2%. On Sokoban, SGPO achieves a success rate of 82.1%, outperforming GRPO and GiGPO. These results indicate consistent improvements over prior group-based baselines across both evaluation metrics.

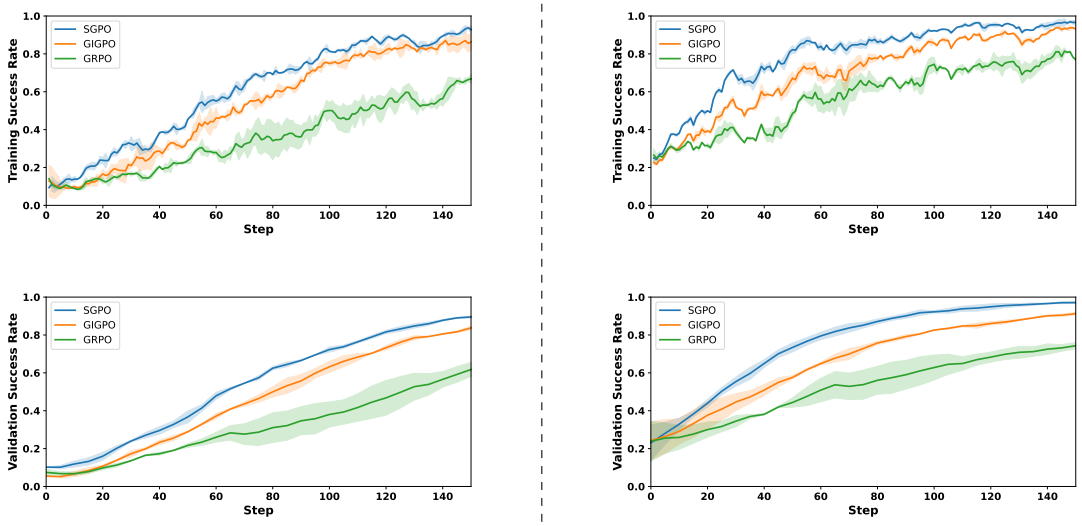


Figure 3: Training and validation success rate curves on ALFWorld for Qwen-2.5-1.5B-Instruct (left) and Qwen-2.5-7B-Instruct (right). Solid lines represent the mean success rate across three random seeds, with shaded areas indicating the standard deviation.

5.3 Ablation Study

To investigate the individual contributions of the hierarchical guidance signals, we conduct an ablation study on ALFWorld using the Qwen2.5-1.5B-Instruct model. We compare the full SGPO framework with two strategic variants: (1) **SGPO w/o inter**, which isolates the effect of local navigation by deactivating the macroscopic progression signal ($\hat{r}_{inter}$); and (2) **SGPO w/o intra**, which focuses on coarse-grained transitions while removing the microscopic guidance ($\hat{r}_{intra}$). The GRPO baseline effectively represents the ablation of all structural coordinates, providing a lower bound to evaluate the cumulative performance gain of our proposed hierarchical structure.

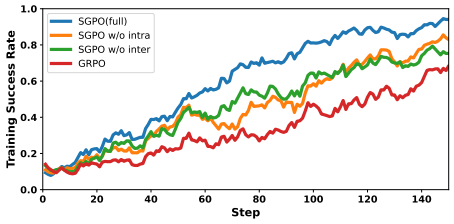


Figure 4: Ablation study on ALFWorld.

The experimental results demonstrate that the full SGPO framework achieves superior performance compared to all variants, validating the synergy between hierarchical signals. The performance drop in SGPO w/o inter indicates that macroscopic guidance is vital for steering the agent through structural bottlenecks toward the global goal. Meanwhile, the gap in SGPO w/o intra suggests that microscopic signals are essential to suppress redundant exploration within local regions. While single-signal variants improve upon the GRPO baseline, their integration is necessary to bridge the gap between local decision-making and long-term task completion.

6 Conclusion

In this paper, we presented Structure-Guided Policy Optimization (SGPO), a framework that improves long-horizon language agent training by grounding credit assignment in recovered task structure. SGPO aggregates self-generated trajectories into an empirical transition graph, abstracts it into topological regions, and derives hierarchical guidance signals for inter-regional progression and intra-regional navigation. These structural signals are integrated with group-based outcome supervision at the advantage level, providing dense step-level guidance. Experiments on ALFWorld, WebShop and Sokoban show that SGPO generally improves performance and training stability over group-based RL baselines, suggesting the value of task-structure-aware optimization for long-horizon agents.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of the manuscript. We have not used generative AI tools for research ideation or execution, retrieval and discovery, or generating synthetic datasets, and other required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. All AI-assisted text was manually reviewed and revised by the authors to ensure its correctness, consistency with the underlying research, and accurate representation of the methods and results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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. *arXiv preprint arXiv:2402.14740*, 2024.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Hyungjoo Chae, Sunghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, et al. Web-shepherd: Advancing prms for reinforcing web agents. *arXiv preprint arXiv:2505.15277*, 2025.

Kevin Chen, Nithin Shrivatsav Srikanth, David Kent, Harish Ravichandar, and Sonia Chernova. Learning hierarchical task networks with preferences from unannotated demonstrations. In *Conference on Robot Learning*, pp. 1572–1581. PMLR, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Sam Michael Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pp. 433–440. IFAAMAS, 2012.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

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

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *Deep Reinforcement Learning Meets Structured Prediction, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Jiazheng Li, Yawei Wang, David Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. Salt: Step-level advantage assignment for long-horizon agents via trajectory graph. *arXiv preprint arXiv:2510.20022*, 2025.

Zihan Lin, Xiaohan Wang, Jie Cao, Jiajun Chai, Guojun Yin, Wei Lin, and Ran He. Rest: Reshaping token-level policy gradients for tool-use large language models. *arXiv preprint arXiv:2509.21826*, 2025.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Online process reward leanring for agentic reinforcement learning. *arXiv e-prints*, pp. arXiv–2509, 2025.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In Carla E. Brodley and Andrea Pohoreckyj Danyluk (eds.), *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pp. 361–368. Morgan Kaufmann, 2001.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Ivan Bratko and Saso Dzeroski (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 278–287. Morgan Kaufmann, 1999.

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

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

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

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

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *arXiv preprint arXiv:2510.14967*, 2025a.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, et al. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. *arXiv preprint arXiv:2510.16907*, 2025b.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023.

Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. Embodied navigation with multi-modal information: A survey from tasks to methodology. *Information Fusion*, 112:102532, 2024.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025.

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

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

Jiazhen Yuan, Wei Zhao, and Zhengbiao Bai. Graph-enhanced policy optimization in llm agent training. *arXiv preprint arXiv:2510.26270*, 2025.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.

A Additional Method Details

A.1 Context-Augmented State Representation

In text-based interactive environments, the raw observations provided by the environment are often incomplete and insufficient to uniquely characterize the agent's underlying task-relevant state. If such incomplete observations are directly used for state canonicalization, different task-relevant contexts may be erroneously mapped to the same state node, leading to a distorted empirical transition graph that misrepresents the true task completion structure.

A core challenge is *state aliasing*, where semantically different contexts yield identical or near-identical observations. For the ALFWorld task *"put a clean butterknife in drawer"*, a successful trajectory completes prerequisite actions (e.g., taking and cleaning the object) before `go to drawer`, whereas an unsuccessful one may execute `go to drawer` immediately. Despite this difference, both produce the same observation upon arriving at the drawer. If canonical states are constructed solely from raw observations, these distinct contexts collapse into a single node, causing the empirical transition graph to suggest a shortcut that bypasses required prerequisites and distorts the recovered task topology.

Similarly, in WebShop, once an agent navigates to a product page, the displayed observation remains invariant regardless of changes to latent configurations such as size, color, or other attributes. Without capturing these updates, all states on the same product page would be merged, obscuring which actions meaningfully advance the purchasing process and which merely induce redundant interactions.

To mitigate state aliasing without introducing privileged supervision, we construct a *context-augmented observation* by supplementing raw inputs with minimal historical context. The sole purpose is to disambiguate observationally identical states; crucially, this augmentation is outcome-agnostic and strictly avoids encoding rewards, success signals, or manual subgoal structures.

Inventory-Based Augmentation in ALFWorld. We augment raw observations with a lightweight inventory to track persistent hidden variables. Specifically, we explicitly record: (i) object acquisition via `take` actions, (ii) object removal via `move` actions, and (iii) property modifications (e.g., `clean`, `heated`, `cooled`) via corresponding interactions. These attributes are attached to the corresponding object identities. This integration distinguishes location-identical states with different task-relevant contexts (e.g., holding a clean object vs. nothing), effectively preventing erroneous state merging.

Action-Conditioned Augmentation in WebShop. In WebShop, purchase-relevant state changes often come from click-induced latent configurations (e.g., selecting size or color) rather than raw page observations. We therefore augment states with a summary of recent configuration-modifying clicks, allowing the transition graph to distinguish different product configurations and separate effective attribute-setting actions from redundant cycles.

Discussion and Limitations. The proposed context-augmented representation is intentionally minimal and does not guarantee perfect state disambiguation. Certain fine-grained dependencies, such as order-sensitive interactions in ALFWorld's *Pick2* tasks, may still induce residual aliasing. Specifically, emptying the inventory after intermediate deposits can cause the state representation to revert to an earlier configuration, creating spurious cycles where distinct sequential contexts collapse into a single node. Nevertheless, empirical results demonstrate that even this conservative augmentation substantially improves the fidelity of the empirical transition graph and supports reliable recovery of task structure.

A.2 Further Discussion on Context Augmentation and State Canonicalization

Evaluating the Role of Context Augmentation. SGPO relies on a reasonable state interface for constructing the empirical transition graph. In partially observable text environments, context augmentation is useful because it reduces state aliasing and allows task-relevant differences to be reflected in canonicalized states. However, this step should be understood as input-level disambiguation rather than benchmark-specific task engineering: it does not encode rewards, success signals, manual subgoals, or predefined stage structures. The SGPO pipeline itself remains unchanged across environments. When observations already provide sufficiently clear state information, such as in Sokoban, no additional context augmentation is required.

To isolate the effect of context augmentation, we conduct an additional ablation on ALFWorld using Qwen2.5-1.5B-Instruct. We compare GiGPO and SGPO under two state interfaces: raw observations and context-augmented observations.

Table 3: Effect of context augmentation on ALFWorld with Qwen2.5-1.5B-Instruct. We report the overall success rate (%).

State Interface	Method	Result (%)
N/A	GRPO	72.8
Raw Observation	GiGPO	86.1
Context Augmentation	GiGPO	85.2
Raw Observation	SGPO	78.9
Context Augmentation	SGPO	89.6

The results show that removing context augmentation leads to a clear performance drop for SGPO, confirming that a reliable state interface is important in partially observable text environments. Nevertheless, even with raw observations, SGPO still achieves 78.9%, outperforming GRPO (72.8%). This indicates that the gain of SGPO is not solely attributable to context augmentation. Instead, the recovered graph structure, region abstraction, and hierarchical guidance still provide additional benefit beyond flat trajectory-level optimization.

Interestingly, equipping GiGPO with the same context augmentation does not improve performance. One possible reason is that GiGPO relies on grouping trajectories by anchor states, while context augmentation makes anchor identities more fine-grained. This can reduce cross-trajectory grouping opportunities and weaken relative comparison. In contrast, SGPO explicitly leverages the resulting connectivity structure, so a more faithful state interface better supports empirical graph construction and structural guidance.

Clarifying the Role of State Canonicalization. State canonicalization is introduced to enable cross-trajectory structural aggregation. A single trajectory only provides one realized execution path, and does not reveal how different rollouts overlap, branch, rejoin, or pass through shared bottlenecks. SGPO is designed to recover task structure from these cross-trajectory regularities. Therefore, comparable observations must be mapped to shared canonical nodes so that empirical connectivity can emerge in the transition graph.

Without canonicalization, the graph would largely degenerate into disconnected or nearly linear trajectory chains, making region abstraction and hierarchical guidance ineffective. In this sense, canonicalized states serve as the entry point for recovering empirical task topology. However, they are an interface-level requirement rather than the core contribution of SGPO. The main contribution lies in how SGPO exploits the recovered topology: it partitions the empirical transition graph into structural regions and derives inter- and intra-regional guidance signals for hierarchical credit assignment.

Context augmentation, similarity-based matching, and raw observation matching can therefore be viewed as different instantiations of the same canonicalization step. Their role is to provide an appropriate notion of state comparability under different environment characteristics. The core SGPO algorithm remains unchanged: it builds an empirical transition graph from sampled trajectories, abstracts the graph into regions, and uses the resulting topology to guide policy optimization. Thus, the need for a reasonable state interface does not reduce SGPO to a task-specific engineering recipe; rather, it reflects the requirement of exposing empirical connectivity before structural credit assignment can be performed.

A.3 Theoretical Analysis of SGPO

We provide a formal characterization of when the structural corrections introduced by SGPO improve step-level credit assignment. Let $A_t^\star$ denote an ideal step-level credit target on the same normalized scale as the trajectory-level outcome advantage $A^O(\tau)$. Since GRPO assigns the same $A^O(\tau)$ to every action within a trajectory, we define the residual step-level credit not captured by this trajectory-level supervision as

$$e_t = A_t^\star - A^O(\tau). \tag{12}$$

SGPO augments the trajectory-level advantage with inter- and intra-regional structural signals:

$$A_t^{\mathrm{SGPO}} = A^O(\tau) + \lambda_{\mathrm{inter}}\hat{r}_{\mathrm{inter},t} + \lambda_{\mathrm{intra}}\hat{r}_{\mathrm{intra},t}. \tag{13}$$

For $k \in \{\mathrm{inter}, \mathrm{intra}\}$, define

$$c_k = \mathbb{E}[e_t \hat{r}_{k,t}], \qquad v_k = \mathbb{E}[\hat{r}_{k,t}^2]. \tag{14}$$

The condition $c_k > 0$ characterizes *structural alignment*: the structural signal is positively aligned with the step-level credit omitted by the flat trajectory-level advantage. In SGPO, the inter-regional signal is positive when a transition reduces the distance toward a success-reachable region and negative when it moves away. Similarly, the intra-regional signal rewards transitions toward a preferred exit while penalizing transitions that move away or induce redundant local exploration. Therefore, when the recovered topology meaningfully reflects task progress, these signals tend to provide positive corrections for useful but under-credited steps and negative corrections for redundant or counterproductive but over-credited steps.

Moreover, the inter-regional signal is applied only to cross-region transitions, whereas the intra-regional signal is applied only to within-region transitions. Their supports are therefore disjoint:

$$\hat{r}_{\mathrm{inter},t}\hat{r}_{\mathrm{intra},t} = 0. \tag{15}$$

The mean-squared credit-assignment error under GRPO is

$$\mathcal{E}_{\mathrm{GRPO}} = \mathbb{E}[e_t^2]. \tag{16}$$

The corresponding error under SGPO can be expanded as

$$\mathcal{E}_{\mathrm{SGPO}} = \mathcal{E}_{\mathrm{GRPO}} + \sum_{k\in\{\mathrm{inter},\mathrm{intra}\}} \left(\lambda_k^2 v_k - 2\lambda_k c_k\right). \tag{17}$$

Hence, for each active structural correction, if $c_k > 0$ and

$$0 < \lambda_k < \frac{2c_k}{v_k}, \tag{18}$$

then

$$\lambda_k^2 v_k - 2\lambda_k c_k < 0, \tag{19}$$

which yields

$$\mathcal{E}_{\mathrm{SGPO}} < \mathcal{E}_{\mathrm{GRPO}}. \tag{20}$$

This result establishes a sufficient condition under which SGPO improves step-level credit assignment: the recovered success-oriented topology should be positively aligned with the step-level contribution omitted by trajectory-level supervision, while the structural coefficients should remain sufficiently controlled. It also indicates that excessively large structural weights may diminish this benefit by amplifying errors in the recovered structure.

A.4 Overall Algorithm

Algorithm 1 summarizes the overall training procedure of SGPO. At each iteration, SGPO first collects a group of trajectories and computes the trajectory-level outcome advantage following GRPO. The same trajectories are then used to construct an empirical transition graph, recover structural regions, and derive inter- and intra-regional guidance signals. These structural signals are normalized and integrated with the outcome advantage for the final policy update.

A.5 Robustness under Success-Scarce Graph Construction

SGPO derives structural guidance from a success-anchored empirical graph. Therefore, the virtual success node and the corresponding success-oriented topological distances are only available when successful trajectories appear in the sampled batch. When no successful trajectory is observed, SGPO does not impose a directional signal on the semantically neutral transition graph. Instead, the

Algorithm 1 Structure-Guided Policy Optimization (SGPO)

Require: Policy π_θ, group size G, $\lambda_{\text{intra}}, \lambda_{\text{inter}}$
for each training iteration **do**
  Sample trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}$
  Compute trajectory-level outcome advantages $A^O(\tau_i)$ via GRPO
  Build empirical transition graph $\mathcal{G}$ from $\mathcal{D}$; add virtual success node v_{succ}
  Partition $\mathcal{G}$ into regions $\mathcal{P} = \{\mathcal{C}_k\}$ and construct $\mathcal{G}_{abs}$
  Identify target region $\mathcal{C}_{\text{target}}$ containing v_{succ}
  Compute inter-regional distances $d_{\text{inter}}(\mathcal{C})$ via reverse BFS on $\mathcal{G}_{abs}$
  for each region $\mathcal{C} \in \mathcal{P}$ **do**
    Compute intra-regional distances $d_{\text{intra}}(s)$ toward preferred exits
  end for
  for each transition $(s_t, a_t, s_{t+1}) \in \mathcal{D}$ **do**
    if $\mathcal{C}(s_t) \neq \mathcal{C}(s_{t+1})$ **then**
      $r_t^{\text{inter}} \leftarrow d_{\text{inter}}(\mathcal{C}(s_t)) - d_{\text{inter}}(\mathcal{C}(s_{t+1}))$
    else
      $r_t^{\text{intra}} \leftarrow d_{\text{intra}}(s_t) - d_{\text{intra}}(s_{t+1})$
    end if
  end for
  Normalize inter-regional signals $\hat{r}_{\text{inter}}$ using global statistics
  Normalize intra-regional signals $\hat{r}_{\text{intra}}$ within each region
  Compute $A^T(a_t, \tau_i) = A^O(\tau_i) + \lambda_{\text{intra}}\hat{r}_{\text{intra}} + \lambda_{\text{inter}}\hat{r}_{\text{inter}}$
  Update π_θ via GRPO objective using A^T
end for

structural terms remain inactive, and the update reduces to standard GRPO using only the trajectory-level outcome advantage. This conservative fallback avoids introducing potentially misleading pseudo-guidance before any successful evidence is available.

This design also clarifies the role of successful trajectories in SGPO. The empirical transition graph is constructed from the full batch, including both successful and failed trajectories. Successful trajectories mainly provide the anchor for orienting the recovered topology toward task completion, while failed trajectories still contribute useful local connectivity, such as repeated states, local branches and redundant cycles. Thus, SGPO does not treat a single successful trajectory as ground-truth task structure; rather, it aggregates connectivity evidence across trajectories and uses successful outcomes to define a direction over the recovered structure.

To evaluate SGPO under success-scarce conditions, we conduct two additional experiments on ALFWorld with Qwen2.5-1.5B-Instruct.

Table 4: Performance under reduced rollout group sizes on ALFWorld with Qwen2.5-1.5B-Instruct. We report success rates (%) at different training steps.

Rollouts	Method	50 Steps	100 Steps	150 Steps
2	GRPO	7.3	11.7	16.6
	GiGPO	12.4	18.1	25.2
	SGPO	11.2	22.2	39.8
4	GRPO	10.8	19.8	28.1
	GiGPO	16.0	40.0	65.0
	SGPO	18.5	47.7	71.1

Reducing the rollout group size. We first reduce the per-task group size from the default $G = 8$ to $G = 4$ and $G = 2$. This makes successful trajectories less likely to appear in each update, thereby weakening the success support available for graph orientation.

As shown in Table 4, SGPO remains effective when the rollout group size is reduced. With $G = 2$, SGPO slightly underperforms GiGPO at 50 steps, but surpasses it at later stages and reaches 39.8% success at 150 steps. With $G = 4$, SGPO consistently outperforms both GRPO and GiGPO across all

reported steps. These results suggest that SGPO can still leverage limited successful experience to construct useful structural guidance, even when each update contains weaker success support.

Table 5: Performance under constrained successful trajectories for graph construction on ALFWorld with Qwen2.5-1.5B-Instruct. We report the final success rate (%).

Method	Graph Construction	Performance (%)
GRPO	–	72.8
GiGPO	–	86.1
SGPO	All successful trajectories	89.6
SGPO	At most 1 successful trajectory	80.5
SGPO	At most 2 successful trajectories	82.8

Constraining successful trajectories in graph construction. We further impose a direct constraint on graph construction. Specifically, when building the empirical graph, we allow at most one or two successful trajectories, together with all failed trajectories. This setting isolates whether SGPO can still recover useful structure when the success-anchored topology is supported by only minimal successful evidence.

Table 5 shows that restricting successful trajectories weakens SGPO, but does not make the method ineffective. Even with at most one successful trajectory for graph construction, SGPO still outperforms GRPO. Increasing the available success support from one to two successful trajectories further improves performance, indicating that richer success yields a more reliable structural orientation.

Overall, these results show that SGPO behaves conservatively when no successful evidence is available, remains useful when success support is limited, and becomes stronger as successful trajectories provide a richer anchor for orienting the recovered task topology.

B Experimental Details and Additional Analysis

B.1 Computational Overhead of SGPO

SGPO introduces additional graph-side operations, including empirical graph construction, topological region partitioning and structural advantage computation. To evaluate the computational cost of these operations, we profile the wall-clock time of SGPO on ALFWorld using Qwen2.5-1.5B-Instruct over 150 training iterations.

SGPO does not maintain a persistent global graph. Instead, it builds an empirical transition graph from the current on-policy batch at each iteration, and performs partitioning and structural computation only on this batch-induced graph. Therefore, the additional cost is per-iteration and batch-bounded, rather than growing with the full theoretical state space.

Table 6: Per-iteration wall-clock time breakdown of SGPO on ALFWorld with Qwen2.5-1.5B-Instruct. We report the average time in seconds over different training stages.

Component	Steps 1–50	Steps 51–100	Steps 101–150	Avg.
Rollout Sampling	434.58	412.70	426.79	424.69
Old-Step Policy Probability Computation	41.98	29.13	19.16	30.09
Reference Probability Computation	40.32	27.13	18.36	28.60
Policy Update	149.34	102.04	68.45	106.61
Graph Construction	9.53	5.83	3.10	6.15
Leiden Clustering	0.03	0.02	0.01	0.02
Structural Advantage Computation	0.01	0.01	0.01	0.01

As shown in Table 6, the additional graph-side overhead is small compared with the dominant components of LLM-agent training, especially rollout sampling and policy update. On average, graph construction takes 6.15 seconds per iteration, while Leiden clustering and structural advantage computation take only 0.02 and 0.01 seconds, respectively. In comparison, rollout sampling alone takes 424.69 seconds per iteration.

The graph-side overhead also decreases over training. This trend is consistent with the structural analysis in Appendix C.1, where the empirical transition graph becomes progressively more compact as the policy concentrates on more stable interaction patterns. As a result, graph construction and subsequent structural computation become cheaper in later training stages. Overall, SGPO adds only a lightweight batch-level structural computation on top of standard GRPO-style training.

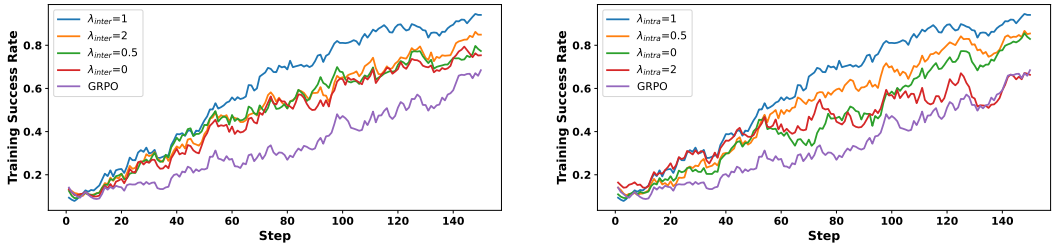
B.2 HYPERPARAMETERS

Table 7 summarizes the key hyperparameter settings used in our experiments. We applied identical configurations across model scales (Qwen-2.5-1.5B and 7B) for both ALFWorld and WebShop, while Sokoban experiments utilized Qwen-2.5-VL-3B.

Table 7: Hyperparameter settings used in our experiments.

Hyperparameter	ALFWorld	WebShop	Sokoban
Mini-batch size	256	64	64
Max interaction steps	50	15	15
Max prompt length	2048	4096	1024
State history length	2	2	0
Rollout temperature	1.0		
Evaluation temperature	0.4		
Group size	8		
Learning rate	1e-6		
Max response length	512		
KL loss coefficient	0.01		
Training steps	150		
Clip ratio	0.2		
Community resolution	1		
Inter-regional guidance coefficient λ_{inter}	1		
Intra-regional guidance coefficient λ_{intra}	1		

B.3 SENSITIVITY ANALYSIS OF STRUCTURAL GUIDANCE WEIGHTS



(a) Effect of λ_{inter} (b) Effect of λ_{intra}

Figure 5: **Sensitivity analysis of structural guidance weights.** We compare the training success rates of Qwen-2.5-1.5B-Instruct on ALFWorld under varying coefficients for (a) macroscopic inter-regional progression and (b) microscopic intra-regional navigation.

We analyze the sensitivity of SGPO to the structural guidance coefficients λ_{inter} and λ_{intra}, which control the strength of macroscopic inter-regional progression and microscopic intra-regional navigation. We evaluate four settings for each coefficient by scaling it to $\{0, 0.5, 1, 2\}$ while keeping all other hyperparameters fixed. The standard GRPO algorithm serves as our baseline, effectively representing the regime where all structural guidance is absent ($\lambda_{\text{inter}} = \lambda_{\text{intra}} = 0$).

Figure 5a illustrates the training success rate under different values of λ_{inter}. Removing the inter-regional signal ($\lambda_{\text{inter}} = 0$) degrades performance relative to non-zero settings. Among non-zero

values, performance exhibits a clear optimum around $\lambda_{\text{inter}} = 1$, while remaining stable without collapse over a broad range (0.5 to 2).

Figure 5b reports the effect of varying λ_{intra}. Disabling intra-regional guidance ($\lambda_{\text{intra}} = 0$) yields a clear degradation relative to optimal non-zero settings, indicating that local navigation signals are important for efficient within-region exploration. A moderate weight ($\lambda_{\text{intra}} = 0.5$) already recovers most of the gain, while the default $\lambda_{\text{intra}} = 1$ achieves the strongest and most stable performance. Increasing the weight further ($\lambda_{\text{intra}} = 2$) introduces noisier training dynamics and slower stabilization, due to over-emphasizing dense intra-region signals, while yielding no clear improvement over GRPO.

Overall, these results demonstrate that both inter- and intra-regional signals are necessary components of SGPO, but the method remains robust within a reasonably wide range of coefficient values. The default setting $\lambda_{\text{inter}} = \lambda_{\text{intra}} = 1$ lies in a stable regime and achieves consistently strong performance.

B.4 Similarity-Based Canonicalization in Open-Ended Search Domains

As discussed in the main method, the state abstraction mapping ϕ serves as a flexible canonicalization interface and can be instantiated by different matching rules. In this subsection, we evaluate a similarity-based instantiation in an open-ended search setting, where observations often differ in surface form even when they play similar interaction roles. This setting tests whether SGPO can recover useful empirical topology when exact state revisitation is sparse, while keeping the rest of the SGPO pipeline unchanged.

Specifically, we replace exact matching with similarity-based matching using Ratcliff–Obershelp similarity. Two states are treated as the same canonical node when their similarity exceeds 0.9. This modification only changes the state interface: the remaining SGPO pipeline, including empirical graph construction, topological region abstraction, and inter-/intra-regional guidance, remains unchanged.

We evaluate this adaptation with Qwen2.5-3B-Instruct in the Search-R1 setting (Jin et al., 2025), where the agent answers questions by interacting with a search engine through tool calls. The benchmark covers both single-hop and multi-hop QA datasets, including in-distribution datasets marked by † and out-of-distribution datasets marked by ⋆.

Table 8: Performance on open-ended search domains. We report results on single-hop and multi-hop QA datasets. † denotes in-distribution datasets and ⋆ denotes out-of-distribution datasets.

Method	NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2Wiki⋆	MuSiQue⋆	Bamboogle⋆	Avg.
Search-R1	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
GiGPO	42.0	59.5	42.4	36.9	37.0	12.6	64.1	42.1
SGPO	44.1	60.8	46.4	36.9	35.7	12.9	62.1	44.1

As shown in Table 8, SGPO improves over both Search-R1 and GiGPO on average. The improvement appears across both single-hop and multi-hop QA tasks, including several out-of-distribution datasets. This suggests that SGPO is not limited to environments where raw observations are exactly repeated. With an appropriate canonicalization interface, structurally similar but non-identical search states can still be merged into shared graph nodes, allowing topological partitioning and hierarchical guidance to remain useful.

B.5 Prompt Templates

This section provides the prompt templates used in our experiments across different benchmarks. These templates are designed to structure the interaction between the LLM agent and the environment, facilitating step-by-step reasoning and precise action selection. The placeholders (e.g., {`current_observation`}, {`admissible_actions`}) are dynamically updated during each interaction step.

ALFWORLD TEMPLATE

```
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} steps. 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.
```

WEBSHOP TEMPLATE

```
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} steps. 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: [{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.
```

SOKOBAN VISUAL TEMPLATE

```
You are an expert agent operating in the Sokoban environment. Your goal
is to push all the boxes onto the target spots. Once all boxes are on
the targets, you win!

# Rules
You can only push boxes. You can't pull them, so plan ahead to avoid
getting stuck. You can't walk through or push boxes into walls. To
avoid traps, do not push boxes into corners or against walls where
they can't be moved again.

# Visual Elements in the Image:
Character: A small, green alien-like figure with two antennae and
black eyes. It is you.
Box: A yellow crate marked with an orange "X" across its front.
It is the box you need to push.
Target: A black tile outlined in red, with a small red diamond
shape in the center. It marks the destination where a box should
be pushed.

# Current Step
Your current observation is shown in the image: <image>
Your admissible actions are ["up", "down", "left", "right"].

Now it's your turn to make a move (choose ONE action only for the
current step).You should first reason step-by-step about the current
situation | observe the positions of boxes and targets, plan a path
to push a box toward a target,and avoid traps like corners or walls.
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.
```

C Structural Analysis and Case Studies

C.1 Evolution of Region Count and State Coverage

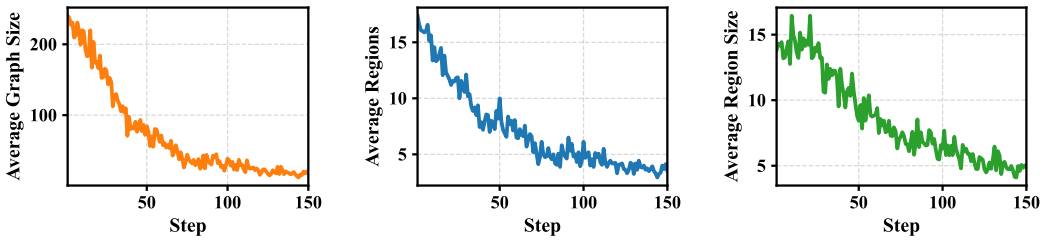


Figure 6: **Topological evolution of the empirical transition graph during training.** The average graph size (left), number of regions (middle) and average region size (right) consistently decrease, reflecting a progressive consolidation of the agent's empirical interaction structure during training.

In this section, we examine the evolution of the empirical transition graph during training, focusing on three structural statistics: the average graph size, the number of recovered regions and the average region size for Qwen2.5-1.5B-Instruct on ALFWorld. As shown in Figure 6, all three metrics exhibit a clear decreasing trend over training, reflecting a progressive contraction of the agent's experienced state space.

The Average Graph Size (Left) decreases sharply from approximately 240 nodes to around 20. This reduction suggests that, as training proceeds, the agent's interaction distribution concentrates on a smaller subset of frequently visited states in the empirical transition graph. Instead of continuing to expand coverage into many distinct states, the learned policy increasingly revisits a compact portion of the graph that repeatedly appears along successful trajectories.

The Average Number of Regions (Middle) declines from roughly 17 to about 4. This trend suggests that the empirical connectivity pattern becomes progressively less fragmented over training. Early on, diverse and inconsistent behaviors produce scattered transitions, leading to many small regions under modularity-based partitioning. As behavior becomes more consistent, these fragments consolidate into a small number of stable regions that capture the dominant structural patterns observed in successful trajectories.

The Average Region Size (Right) contracts from approximately 14 to around 5 states. This reflects a reduction in local behavioral variability within each region. As training progresses, transitions within a region become more concentrated around a limited set of states, indicating that the agent follows increasingly consistent local transition patterns rather than exploring a wider range of local variations.

C.2 Structural Comparison Between GiGPO and SGPO

To further clarify the structural advantage of SGPO over GiGPO, we provide a qualitative comparison using the same sampled trajectories. GiGPO performs relative evaluation around shared anchor states, whereas SGPO aggregates trajectories into an empirical transition graph and recovers region-level task topology for hierarchical credit assignment.

Figure 8a illustrates the GiGPO view. GiGPO groups trajectory fragments associated with the same anchor state and computes relative advantages within each group. This anchor-centered comparison is useful for local credit assignment, since actions around the same anchor can be evaluated under a comparable context. However, the resulting structure remains local. It does not explicitly reveal latent task stages, bottleneck transitions, or the global organization of states with respect to task completion.

This locality can limit credit assignment in long-horizon tasks. As shown in Figure 8b(left), even when actions share the same anchor state, redundant or suboptimal downstream continuations may still affect the resulting relative advantage, coupling delayed trajectory-level effects back to earlier decisions. In addition, some anchor states may appear only once in the sampled batch, yielding degenerate comparison groups with little discriminative signal, as shown in Figure 8b(right).

Figure 9 shows the SGPO view on the same trajectories. SGPO first aggregates all sampled transitions into an empirical transition graph and partitions it into structural regions based on connectivity patterns. Densely connected states are absorbed into the same region, while sparse cross-region transitions expose higher-level progression structure. This region-level abstraction makes the latent stage organization explicit: intra-region transitions capture local variation, whereas inter-region transitions correspond to structural progress.

The recovered topology is then directly used to construct hierarchical guidance signals. Region-level connectivity defines inter-regional progression and the success-oriented topological distance, while local connectivity within each region identifies preferred exits for intra-regional navigation. Thus, SGPO not only produces a more structured representation, but also converts this topology into macroscopic progression signals and microscopic navigation signals.

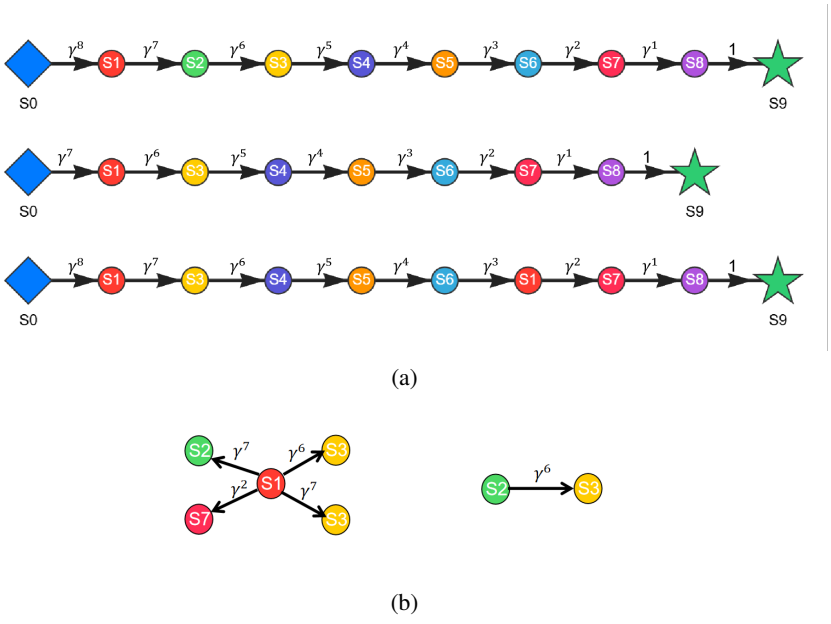
Node List (State)

Node 0: -= Welcome to TextWorld, ALFRED! You are in the middle of a room.Looking quickly around you,you see a cabinet 13,a cabinet 12....
Node 1:You arrive at countertop 1.On the countertop 1,you see a cup 1,a houseplant 1,a mug 1,and a peppershaker 1.
Node 2:You arrive at stoveburner 1.On the stoveburner 1,you see a pot 2.
Node 3:You arrive at countertop 2.On the countertop 2,you see a butterknife 1,a cellphone 1,a egg 1,a fork 2,a fork 1,a knife 2,a knife 1,a ladle 1,a lettuce 1,and a tomato 1.
Node 4:You pick up the butterknife 1 from the countertop 2.
Node 5:You arrive at sinkbasin 1.On the sinkbasin 1,you see a spatula 1,a spoon 2,and a spoon 1. Inventory:butterknife.
Node 6:You clean the butterknife 1 using the sinkbasin 1.Inventory:butterknife.
Node 7:You arrive at drawer 1.The drawer 1 is closed.Inventory:cleaned butterknife.
Node 8:You open the drawer 1.The drawer 1 is open.In it,you see nothing.Inventory:cleaned butterknife.
Node 9:Task completed successfully!
Node 10:You arrive at countertop 1.On the countertop 1,you see a cup 1,a houseplant 1,a mug 1,and a peppershaker 1.Inventory:cleaned butterknife.

Edge List (Action)

Node 0 → Node 1:go to countertop 1
Node 1 → Node 2:go to stoveburner 1
Node 2 → Node 3:go to countertop 2
Node 3 → Node 4:take butterknife 1 from countertop2
Node 4 → Node 5:go to sinkbasin 1
Node 5 → Node 6:clean butterknife 1 with sinkbasin1
Node 6 → Node 7:go to drawer 1
Node 7 → Node 8:open drawer 1
Node 8 → Node 9:move butterknife 1 to drawer 1
Node 1 → Node 3:go to countertop 2
Node 6 → Node 10:go to countertop 1
Node 10 → Node 7:go to drawer 1

Figure 7: The text description of nodes and edges of the state graph in Fig 8, Fig 9. The highlighted red context-augmented states are only used in original SGPO (Fig 9).



(a)

(b)

Figure 8: **Illustration of GiGPO.** (a) Multiple sampled trajectories may share the same anchor states under different local contexts. (b) GiGPO performs relative credit assignment by grouping trajectory fragments around a shared anchor state. This anchor-centered mechanism supports local comparison, but does not explicitly recover a global task topology.

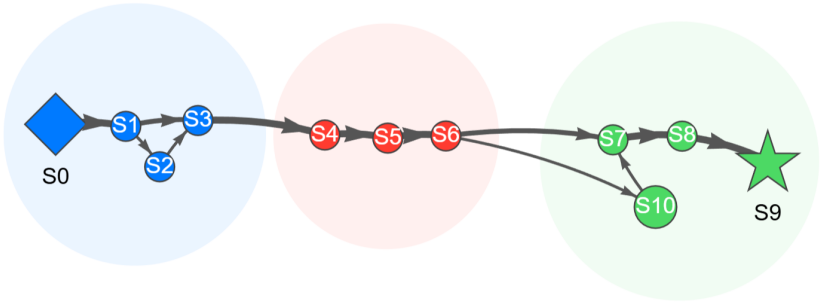


Figure 9: **SGPO on the same trajectories.** SGPO aggregates the sampled trajectories into an empirical transition graph and partitions the graph into structural regions. This region-level abstraction exposes global task progression and bottleneck transitions, providing an explicit topological coordinate beyond anchor-based local comparison.

C.3 Visualization of Structural Regions (Case Examples)

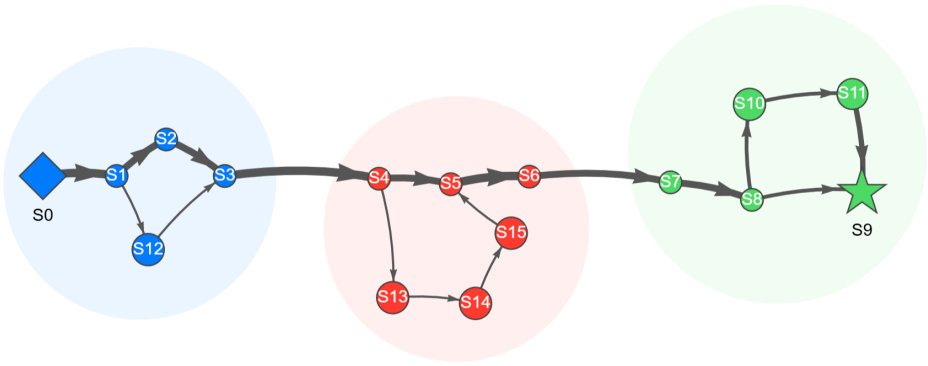


Figure 10: **Visualization of ALFWorld task topology and regional abstraction.** This graph, built from 8 trajectories, shows how SGPO clusters fine-grained states into distinct structural regions. The blue diamond and green star indicate the start (S_0) and the success sink (v_{succ}). Wider edges represent more frequent transitions. This abstraction transforms a flat sequence into a hierarchical coordinate system for guidance.

Node List

Node 0: -= Welcome to TextWorld, ALFRED! =-
You are in the middle of a room.Looking quickly around you,you see a cabinet 13,a cabinet 12....
Node 1:You arrive at countertop 1.On the countertop 1,you see a cup 1,a houseplant 1,a mug 1,and a peppershaker 1.
Node 2:You arrive at stoveburner 1.On the stoveburner 1,you see a pot 2.
Node 3:You arrive at countertop 2.On the countertop 2,you see a butterknife 1,a cellphone 1,a egg 1,a fork 2,a fork 1,a knife 2,a knife 1,a ladle 1,a lettuce 1,and a tomato 1.
Node 4:You pick up the butterknife 1 from the countertop 2.
Node 5:You arrive at sinkbasin 1.On the sinkbasin 1,you see a spatula 1,a spoon 2,and a spoon 1. Inventory:butterknife.
Node 6:You clean the butterknife 1 using the sinkbasin 1.Inventory:butterknife.
Node 7:You arrive at drawer 1.The drawer 1 is closed.Inventory:cleaned butterknife.
Node 8:You open the drawer 1.The drawer 1 is open.In it,you see nothing.Inventory:cleaned butterknife.
Node 9:Task completed successfully!
Node 10:You arrive at drawer 2.The drawer 2 is closed.Inventory:cleaned butterknife.
Node 11:You open the drawer 2.The drawer 2 is open.In it,you see nothing.Inventory:cleaned butterknife.
Node 12:You arrive at fridge 1.The fridge 1 is closed.
Node 13:You arrive at drawer 2.The drawer 2 is closed.Inventory:butterknife.
Node 14:You open the drawer 2.The drawer 2 is open.In it,you see nothing.Inventory:butterknife.
Node 15:Nothing happens.Inventory:butterknife.

Edge List

Node 0 → Node 1:go to countertop 1
Node 1 → Node 2:go to stoveburner 1
Node 2 → Node 3:go to countertop 2
Node 3 → Node 4:take butterknife 1 from countertop2
Node 4 → Node 5:go to sinkbasin 1
Node 5 → Node 6:clean butterknife 1 with sinkbasin1
Node 6 → Node 7:go to drawer 1
Node 7 → Node 8:open drawer 1
Node 8 → Node 9:move butterknife 1 to drawer 1
Node 1 → Node 12:go to fridge 1
Node 12 → Node 3:go to countertop 2
Node 4 → Node 13:go to drawer 2
Node 13 → Node 14:open drawer 2
Node 14 → Node 15:clean butterknife 1(invalid)
Node 15 → Node 5:go to sinkbasin 1
Node 8 → Node 10:go to drawer 2
Node 10 → Node 11:open drawer 2
Node 11 → Node 9:move butterknife 1 to drawer 2

Figure 11: The text description of nodes and edges of the state graph in Fig 10.

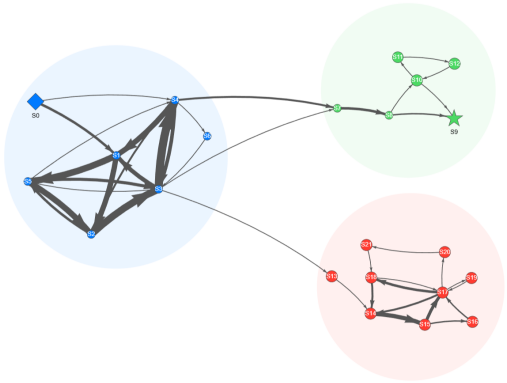


Figure 12: **Visualization of ALFWorld task topology and regional abstraction.** This graph aggregates 8 trajectories, including 2 failed attempts that terminated at the 50-step limit. The blue diamond and green star indicate the start (S_0) and the success sink (v_{succ}). Wider edges represent more frequent transitions. This abstraction transforms a flat sequence into a hierarchical coordinate system for guidance.

Node List

Node 0: -= Welcome to TextWorld, ALFRED! You are in the middle of a room.Looking quickly...
Node 1:You arrive at desk 1.On the desk 1,you see a alarmclock 1,a book 1,a desklamp 1,a keychain...
Node 2:You turn on the desklamp 1.
Node 3:You arrive at dresser 1.On the dresser 1,you see a pen 2,and a pencil 2.
Node 4:You arrive at bed 1.On the bed 1,you see a laptop 1,and a pillow 1.
Node 5:Nothing happens.
Node 6:You are facing the bed 1.Next to it,you see nothing.
Node 7:You arrive at garbagecan 1.On the garbagecan 1,you see a cd 1,and a pencil1.
Node 8:You pick up the cd 1 from the garbagecan 1.
Node 9:Task completed successfully!
Node 10:Nothing happens.Inventory:cd.
Node 11:You arrive at dresser 1.On the dresser 1,you see a pen 2,and a pencil 2.Inventory:cd.
Node 12:You arrive at bed 1.On the bed 1,you see a laptop 1,and a pillow 1.Inventory:cd.
Node 13:You pick up the pen 2 from the dresser 1.
Node 14:You arrive at desk 1.On the desk 1,you see a alarmclock 1,a book 1,a desklamp 1,a keychain 1,and a pencil 3.Inventory:pen.
Node 15:You turn on the desklamp 1.Inventory:pen.
Node 16:You are facing the desk 1.Next to it,you see nothing.Inventory:pen.
Node 17:You arrive at dresser 1.On the dresser 1,you see a pencil 2.Inventory:pen.
Node 18:You arrive at bed 1.On the bed 1,you see a laptop 1,and a pillow 1.Inventory:pen.
Node 19:You arrive at garbagecan 1.On the garbagecan 1,you see a cd 1,and a pencil 1.Inventory:pen.
Node 20:You arrive at drawer 1.The drawer 1 is closed.Inventory:pen.
Node 21:You open the drawer 1.The drawer 1 is open.In it,you see a creditcard 1,and a pen 1.Inventory:pen.

Edge List

Node 0 → Node 1:go to desk 1
Node 1 → Node 2:use desklamp 1
Node 2 → Node 3:go to dresser 1
Node 3 → Node 4:go to bed 1
Node 5 → Node 4:go to bed 1
Node 4 → Node 6:look
Node 6 → Node 3:go to dresser 1
Node 4 → Node 3:go to dresser 1
Node 1→Node 5:take desklamp 1 from desk 1(invalid)
Node 2 → Node 4:go to bed 1
Node 0 → Node 4:go to bed 1
Node 3 → Node 1:go to desk 1
Node 4 → Node 1:go to desk 1
Node 2 → Node 5:look at cd under desklamp(invalid)
Node 5 → Node 2:use desklamp 1
Node 4 → Node 7:go to garbagecan 1
Node 3 → Node 7:go to garbagecan 1
Node 7 → Node 8:take cd 1 from garbagecan 1
Node 8 → Node 9:go to desk 1
Node8→Node10:look at cd 1 under desklamp 1(invalid)
Node 10 → Node 9:go to desk 1
Node 10 → Node 11:go to dresser 1
Node 11 → Node 12:go to bed 1
Node 12 → Node 10:look at cd 1(invalid)
Node 3 → Node 13:take pen 2 from dresser 1
Node 13 → Node 14:go to desk 1
Node 14 → Node 15:use desklamp 1
Node 15 → Node 16:look
Node 16 → Node 17:go to dresser 1
Node 17 → Node 18:go to bed 1
Node 18 → Node 17:go to dresser 1
Node 17 → Node 14:go to desk 1
Node 15 → Node 17:use desklamp 1
Node 17 → Node 19:go to garbagecan 1
Node 19 → Node 17:go to dresser 1
Node 17 → Node 20:open drawer 1
Node 20 → Node 21:go to bed 1
Node 18 → Node 14:go to desk 1
Node 21 → Node 18:go to bed 1

Figure 13: The text description of nodes and edges of the state graph in Fig 12.

C.4 CASE STUDY

C.4.1 ALFWORLD

Step 1

```
Env(FULL):
You are an expert agent operating in the ALFRED Embodied Environment.
Your current observation is: -= Welcome to TextWorld, ALFRED! =-

You are in the middle of a room. Looking quickly around you, you see a
cabinet 13,a cabinet 12, a cabinet 11, a cabinet 10, a cabinet 9, a
cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a
cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop
4, a countertop 3, a countertop 2, a countertop 1, a drawer 27, a
drawer 26, a drawer 25, a drawer 24, a drawer 23, a drawer 22, a drawer
21, a drawer 20, a drawer 19, a drawer 18, a drawer 17, a drawer 16, a
drawer 15, a drawer 14, a drawer 13,a drawer 12, a drawer 11, a drawer
10, a drawer 9, a drawer 8, a drawer 7, a drawer 6, a drawer 5, a drawer
4, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a
microwave 1, a sinkbasin 1, a stoveb-urner 4, a stoveburner 3, a
stoveburner 2, a stoveburner 1, and a toaster 1.

Your task is to: put a clean butterknife in drawer.
Your admissible actions of the current situation are: ['go to cabinet 1'
 'go to cabinet 10'
           ......

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.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. First,I need to
find a butterknife, which could be in any of the cabinets, countertops,
or other locations. I will start by examining the countertops and
cabinets since they are the most likely places to find a butterknife.
</think>
<action>
go to countertop 1
</action>
```

Step 2

```
Env(Observation):
You arrive at countertop 1. On the countertop 1, you see a cup 1, a
houseplant 1,a mug 1, and a peppershaker 1.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. From the current
observation, I see no butterknife on the countertop 1. The countertop 1
contains a cup 1, a houseplant 1, a mug 1, and a peppershaker 1. I need
to search for a butterknife first. The locations I have visited so far
are countertop 1. I should explore othercountertops and cabinets to find
the butterknife.
</think>
<action>
go to countertop 2
</action>
```

Step 3

```
Env(Observation):
You arrive at countertop 2. On the countertop 2,you see a butterknife 1,
a cellphone 1, a egg 1, a fork 2, a fork 1, a knife 2, a knife 1, a
ladle 1, a lettuce 1,and a tomato 1.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. From the current
observation, I see a butterknife 1 on the countertop 2. I need to take
the butterknife,clean it,and then put it in the drawer. The first step
is to take the butterknife from the countertop 2.
</think>
<action>
take butterknife 1 from countertop 2
</action>
```

Step 4

```
Env(Observation):
You pick up the butterknife 1 from the countertop 2.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. I currently have a
butterknife 1 in my inventory. I need to clean it first before I can put
it in the drawer.The most logical action to take is to go to the
sinkbasin 1,as it is a common place to wash kitchen items. I will go
there and use the sinkbasin to clean the butterknife.
</think>
<action>
go to sinkbasin 1
</action>
```

Step 5

```
Env(Observation):
You arrive at sinkbasin 1.On the sinkbasin 1,you see a spatula 1,a spoon
2,and a spoon 1.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. I currently have a
butterknife 1 in my inventory and I am at the sinkbasin 1. The first step
is to clean the butterknife. The action 'clean butterknife 1 with
sinkbasin 1' is available,so I will use it to clean the butterknife.
</think>
<action>
clean butterknife 1 with sinkbasin 1
</action>
```

Step 6

```
Env(Observation):
You clean the butterknife 1 using the sinkbasin 1.
------------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. I have already
cleaned the butterknife 1 in the sinkbasin 1. Now, I need to find a
butterknife in the drawer and put the clean one I have in the drawer.
From the current observation, I see that I am in the sinkbasin 1 and I
have a clean butterknife 1 in my inventory. I need to go to the drawer
to put the butter-knife in it. The action 'go to drawer 1' is available,
```

```
so I will use it.
</think>
<action>
go to drawer 1
</action>
```

Step 7

```
Env(Observation):
You arrive at drawer 1. The drawer 1 is closed.
----------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. From the current
observation, I see that the drawer 1 is closed. I need to open it to put
the butterknife inside.The action 'open drawer 1' is available,so I will
use it to open the drawer and put the butterknife inside.
</think>
<action>
open drawer 1
</action>
```

Step 8

```
Env(Observation):
You open the drawer 1. The drawer 1 is open. In it, you see nothing.
----------------------------------------------------------------------
Agent:
<think>
The task is to put a clean butterknife in the drawer. From the current
observation,I see that the drawer 1 is open and contains no items.I also
know that I have cleaned a butterknife 1 in the sinkbasin 1.The next step
is to place the clean butterknife in the open drawer. The action 'move
butterknife 1 to drawer 1' is available, which is the correct action to
perform this task.
</think>
<action>
move butterknife 1 to drawer 1
</action>
```

C.4.2 WEBSHOP

Step1

```
Env(FULL):
You are an expert autonomous agent operating in the WebShop e-commerce
environment. Your task is to: Find me sconces with clear glass, glass
shade, vanity light, light fixture for living room, dining room with
color: 1-light, and size: 2-pack, and price lower than 130.00 dollars.
Your current observation is: 'Search'.
Your admissible actions of the current situation are:
[
'search[<your query>]',
'click[search]',
].

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.
----------------------------------------------------------------------
Agent:
```

```
<think>
The task is to find sconces with specific criteria: clear glass,glass
shade,vanity light,light fixture for living room, dining room, color
1-light, size 2-pack, and price lower than $130.00. The current
observation is 'Search',meaning that the agent needs to submit a search
query. The search query should include all the required attributes and
filters to narrow down the search results to the appropriate products.
The query needs to cover the following points:
- Sconces
- Clear glass
- Glass shade
- Vanity light
- Light fixture
- Living room
- Dining room
- Color 1-light
- Size 2-pack
- Price below $130.00
A well-constructed search query will lead to more relevant results,
making it easier to filter and select the correct items.
</think>
<action>
search[sconces clear glass glass shade vanity light light fixture
living room dining room color 1-light size 2-pack price < 130.00]
</action>
```

Step2

```
Env(Observation):
'Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] 'Next >' [SEP]
'B0928LGTVF' [SEP] 'Fanyate Antique Industrial Wall Sconce, 2-Light
Bathroom Light Fixture Oil Rubbed Bronze Vanity Light with Clear Glass
Shade Suitable for Bathroom LivingRoom Hallway ORB,2 Pack' [SEP]'$113.99'
[SEP] 'B095SWN6C3' [SEP] '1-Light Farmhouse Wall Sconces Black Bathroom
Vanity Light with Glass Shade, Bathroom LightFixture Over Mirror,Modern
Indoor Wall Light for Bedroom Living Room Hallway Porch'[SEP] '$36.99'
[SEP] 'B08D74SKT2' [SEP] '4-Light Wall Sconces Vintage Bathroom Vanity
Light Over Mirror in Brushed Nickel Finish' [SEP] '$119.0' [SEP]
'B0854JTZMT'[SEP]'BAYCHEER Tiffany Style Stained Glass Bowl Shade Pendant
Light Chandelier Decorative Hanging Lamp Pendant Lighting Adjustable
Ceiling Fixture with Pull Chain 3 Lights for Living Room Dining Room'
[SEP] '$169.99' [SEP] 'B07ZBCF88X' [SEP] 'VINLUZ Farmhouse Chandeliers
Rectangle Black 5 Light Dining RoomLighting Fixtures Hanging, Kitchen
Island Cage Pendant Lights Contemporary Modern Ceiling Light with Glass
Shade Adjustable Rods' [SEP] '$199.99' [SEP] 'B07Q87P8DQ' [SEP] 'Permo
Vintage Rustic Industrial 3-Lights Kitchen Island Chandelier Triple 3
Heads Pendant Hanging Ceiling Lighting Fixture with Oval Cone Clear
Glass Shade(Antique)'[SEP] '$94.99' [SEP] 'B08DHS3TH5' [SEP] 'LALUZ
Bathroom Light Fixtures, Gold Vanity Light Fixture with Clear Glass
Shades ( L19.5"× W6"× H7.5")' [SEP]'$117.99' [SEP] 'B01NCQLFPH' [SEP]
'Designers Impressions Juno Oil Rubbed Bronze 2 Light WallSconce/Bathroom
Fixture with Clear and Frosted Glass: 73470' [SEP] '$55.95' [SEP]
'B09J9Y9H95' [SEP] 'Kira Home Ainsley 21.5" 3-Light Farmhouse
Vanity/Bathroom Light+ Clear Cylinder Glass Shades, Oil-Rubbed Bronze
Finish' [SEP] '$77.99' [SEP] 'B003HBR86S' [SEP] 'Sea Gull Lighting
85200-12 Wynfield One-Light Outdoor Wall Lantern
with Clear Beveled Glass Panels, Black Finish' [SEP] '$61.17'.
--------------------------------------------------------------------------
Agent:
<think>
The task is to find sconces with clear glass, glass shade, vanity light,
light fixture for living room, dining room, color 1-light, size 2-pack,
and price lower than $130.00.The current observation shows a list of
products and their prices.The first product, B0928LGTVF, seems to match
```

```
the criteria as it has "2 Pack" mentioned, and the price is $113.99,
which is below $130.00. It also mentions "2-Light Bathroom Light
Fixture" which could fit the living room or dining room context. Let's
click on this product to check its details.
</think>
<action>
click[b0928lgtvf]
</action>
```

Step3

```
Env(Observation):
'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] '1-light' [SEP]
'1-light-gd'[SEP]'2-light' [SEP] '2-light-gd' [SEP] '3-light-gd' [SEP]
'size' [SEP] '1-pack' [SEP] '2-pack' [SEP] 'Fanyate Antique Industrial
Wall Sconce, 2-Light Bathroom Light Fixture Oil Rubbed Bronze Vanity
Light with Clear Glass Shade Suitable for Bathroom Living Room Hallway
ORB, 2 Pack' [SEP] 'Price: $113.99' [SEP] 'Rating: N.A.' [SEP]
'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.
----------------------------------------------------------------------
Agent:
<think>
The task is to find sconces with clear glass, glass shade, vanity light,
light fixture for living room, dining room with color: 1-light,and size:
2-pack, and price lower than 130.00 dollars.
From the previous observations,I have been searching for the appropriate
product and clicking on the most relevant one, which was B0928LGTVF.
However, it did not meet the criteria as it was 2-light, not 1-light,and
it was priced at $113.99,which is within the budget but does not match
the color requirement.Given the current observation, the product is
Fanyate Antique Industrial Wall Sconce, 2-Light Bathroom Light Fixture
Oil Rubbed Bronze Vanity Light with Clear Glass Shade. It meets the size
requirement (2-pack) but not the color requirement(1-light).I will click
on the "1-light" color option to see if it is available. If not, I will
consider other options.
</think>
<action>
click[1-light]
</action>
```

Step4

```
Env(Observation):
'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] '1-light' [SEP]
'1-light-gd' [SEP]'2-light' [SEP] '2-light-gd' [SEP] '3-light-gd' [SEP]
'size' [SEP] '1-pack' [SEP] '2-pack' [SEP] 'Fanyate Antique Industrial
Wall Sconce, 2-Light Bathroom Light Fixture Oil Rubbed Bronze Vanity
Light with Clear Glass Shade Suitable for Bathroom Living Room Hallway
ORB, 2 Pack' [SEP] 'Price: $113.99' [SEP] 'Rating: N.A.' [SEP]
'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.
----------------------------------------------------------------------
Agent:
<think>
The task is to find sconces with clear glass, glass shade, vanity light,
light fixture for living room, dining room with color:1-light,and size:
2-pack, and price lower than 130.00 dollars.
From the previous observations and actions,I have been navigating through
the search results,clicking on specific products,and filtering the color
to 1-light. However, I haven't found any product that meets all the
criteria yet. The last product I clicked on was the Fanyate Antique
Industrial Wall Sconce, which has a 1-light color but does not meet the
size requirement as it is 2-pack,and it is also within the budge at
$113.99.Given the current observation, the next step is to filter the
size to 2-pack. This will narrow down the search results to products
```

```
that fit the size requirement.
</think>
<action>
click[2-pack]
</action>
```

Step5

```
Env(Observation):
'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] '1-light' [SEP]
'1-light-gd' [SEP]'2-light' [SEP] '2-light-gd' [SEP] '3-light-gd' [SEP]
'size' [SEP] '1-pack' [SEP] '2-pack' [SEP] 'Fanyate Antique Industrial
Wall Sconce, 2-Light Bathroom Light Fixture Oil Rubbed Bronze Vanity
Light with Clear Glass Shade Suitable for Bathroom Living Room Hallway
ORB, 2 Pack' [SEP] 'Price: $113.99' [SEP] 'Rating: N.A.' [SEP]
'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.
----------------------------------------------------------------------
Agent:
<think>
The task is to find sconces with clear glass, glass shade, vanity light,
light fixture for living room,dining room with color: 1-light,and size:
2-pack, and price lower than 130.00 dollars.
From the previous observations and actions,I have been navigating
through the search results and clicking on specific options (color and
size) to narrow down the search. The last action I performed was to
click on the "2-pack" size, which led me to the product description of
the Fanyate Antique Industrial Wall Sconce,2-Light Bathroom Light
Fixture.This product meets most of the criteria, except that its price
is $113.99,which is below the specified limit.Therefore,I can proceed
to buy this product as it fits all the requirements.I will now click on
the "Buy Now" button to purchase the item.
</think>
<action>
click[buy now]
</action>
```

C.4.3 SOKOBAN

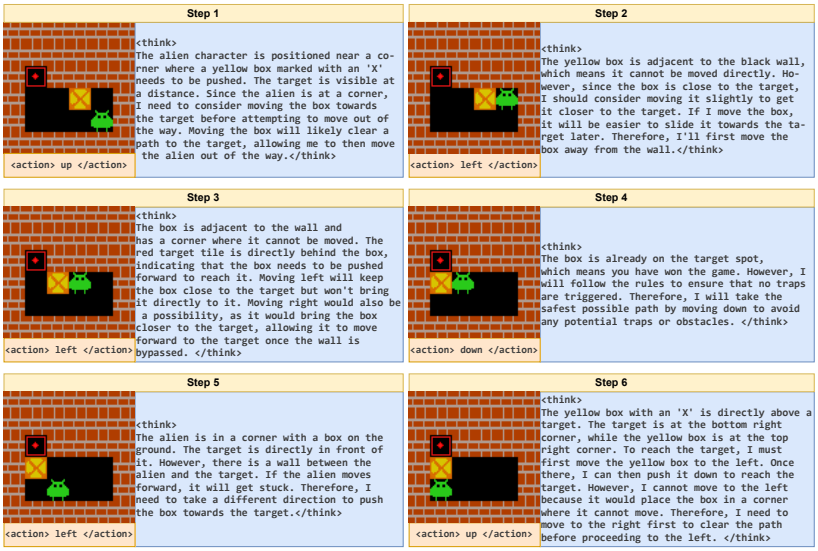


Figure 14: A Case Study of Sokoban