EXPLICIT TRAJECTORY DIVERSITY FOR RL-BASED POST-TRAINING OF LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents often admit multiple high-quality solutions to the same task, differing in reasoning structure, tool-use pattern, or interaction trajectory. Yet existing notions of diversity in LLM post-training are mostly implicit, arising from general stochasticity and regularization mechanisms rather than explicitly targeting task-relevant behavioral variation. While such implicit diversity can be useful, it does not directly specify which forms of behavioral variation should be encouraged for a given task. In this work, we study *explicit trajectory diversity* in RL-based post-training for LLMs. Our key idea is to define diversity through user-specified, task-specific trajectory descriptors, which map each sampled trajectory to an interpretable behavioral representation, and then measure diversity as a set-level functional over the resulting descriptor matrix. Building on this formulation, we introduce *Trajectory-guided Joint Policy Optimization* (TJPO), a single-policy framework that optimizes explicit diversity over sampled trajectory groups, avoiding the need for population-based policy training, and instantiate it within group-based policy optimization through trajectory-level learning signals. This design makes the diversity objective both interpretable and controllable: practitioners can specify whether diversity should correspond to action composition, location visitation patterns, spatial coverage, or other task-relevant behavioral dimensions. Experiments on Sokoban and ALFWorld show that TJPO improves task-specific trajectory diversity while maintaining competitive task performance. Descriptor and trajectory analyses show that the learned variation follows the specified behavioral dimensions and includes distinct successful strategies. Beyond diversity metrics, evaluations on an ALFWorld subset with verified alternative solutions demonstrate practical benefits without additional training: compared with GiGPO, TJPO improves compliant success under user-specified location constraints by 8.3 percentage points and task success under environmental disruptions by 7.5 percentage points. These findings suggest that explicitly shaping trajectory diversity can help LLM agents satisfy user requirements and remain effective when task conditions change.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as agents that solve multi-step tasks through planning, interaction, and tool use (Du et al., 2026; Zhang et al., 2026). In such settings, a single objective often admits multiple high-quality solutions that differ substantially in their behavioral patterns, such as plan decomposition, tool selection, and interaction trajectories (Yao et al., 2022; Shridhar et al., 2021; Schrader, 2018). For example, consider a household agent asked to place a target object in a desired location. One successful strategy may first explore the environment to identify relevant objects and receptacles, verify object states, and then execute the manipulation sequence. Another strategy may directly exploit known task regularities, navigate to likely object locations, and perform a shorter sequence of actions. Both strategies can solve the task, but they differ in object-interaction patterns, action composition, and robustness to changes in the environment. Similarly, in a planning game such as Sokoban, two successful solutions may reach the same goal state while following different routes, pushing boxes in different orders, or using different intermediate subgoals. Therefore, beyond average success rate or expected return, practical deployment also benefits from sufficient behavioral coverage: an agent should be able to produce a set of complemen-

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.

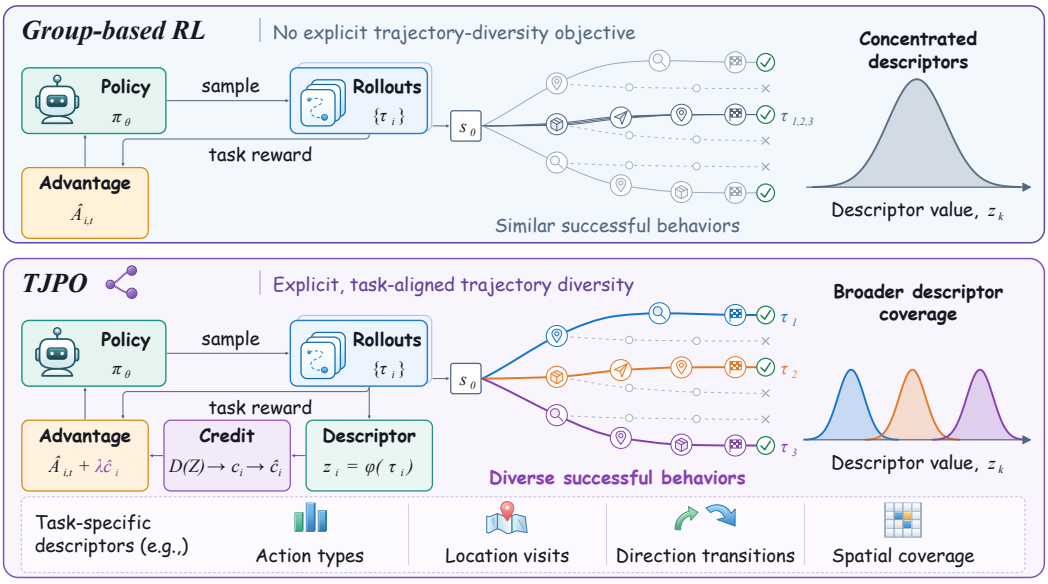


Figure 1: Motivation and schematic overview of TJPO. Without an explicit diversity objective, successful rollouts may concentrate on similar behaviors (top). TJPO augments the group-based advantage with descriptor-based trajectory-level diversity credits to encourage task-relevant behavioral variation (bottom).

tary high-quality strategies rather than relying on a single dominant path (Batra et al., 2023; Cully & Demiris, 2017). Such coverage is useful because the preferred strategy may depend on deployment constraints. A conservative retrieve-then-verify workflow may be preferable when reliability is critical, while a more direct execution-oriented strategy may be preferable when latency or interaction cost is limited. Moreover, when the environment changes (e.g., when an object is unavailable, a tool fails, or the accessible action set is restricted), an agent with multiple behaviorally distinct strategies is more likely to retain a viable alternative path. This motivates diversity not merely as variation in final responses, but as task-relevant variation in the trajectories by which agents achieve their goals.

However, such behavioral coverage does not follow automatically from RL-based post-training, where optimization primarily targets task rewards or preference alignment (Bai et al., 2024; Feng et al., 2025a; Chen et al., 2025a; Wang et al., 2025; Feng et al., 2025b). Without incentives to preserve diversity, policy optimization can favor a small set of reliably rewarded behaviors, while other valid strategies become less likely to be sampled and reinforced (Kirk et al., 2024). Existing mechanisms can encourage diversity through decoding heuristics, entropy regularization, or other distribution-level regularizers (Yao et al., 2025; Li et al., 2025a). While these mechanisms can support exploration and response variation, encouraging variation alone does not specify *what* should be diverse for a particular task. These mechanisms typically operate on generic properties of the policy distribution rather than task-specific behavioral dimensions, leaving their effect on strategy-level diversity implicit. Consequently, greater response variation does not necessarily translate into distinct reasoning structures, planning patterns, or tool-use behaviors that matter in practice. The resulting diversity may therefore remain weakly aligned with task requirements and difficult to interpret or control. This motivates a key question: how can we explicitly define and optimize task-relevant trajectory diversity in RL-based post-training while maintaining strong task performance?

To address this question, we propose *Trajectory-guided Joint Policy Optimization* (TJPO), a simple framework for optimizing user-specified, task-specific trajectory diversity in RL-based post-training. Our key idea is to represent each sampled trajectory using a set of trajectory descriptors chosen to reflect the behavioral variations that matter for the target environment, and to define diversity as a set-level functional over the resulting descriptor matrix. This formulation makes the diversity objective explicit and controllable, rather than leaving diversity to emerge indirectly from generic stochasticity or response variation. We further integrate this signal into group-based policy optimiza-

tion by translating group-level diversity into trajectory-level credit, making TJPO readily compatible with GRPO-style and GiGPO-style training. Figure 1 illustrates the motivation and diversity credit introduced by TJPO.

We evaluate TJPO on two complementary environments: Sokoban, a planning-intensive game environment, and ALFWorld, a long-horizon agentic environment (Shridhar et al., 2021; Schrader, 2018). For each environment, we instantiate user-specified, task-specific trajectory descriptors tailored to the corresponding notion of meaningful behavioral variation. On both environments, TJPO improves explicit trajectory diversity over strong group-based RL baselines while maintaining competitive task performance. These results suggest that task-specific trajectory descriptors provide a practical interface for optimizing meaningful behavioral diversity in RL-based post-training. Our contributions are summarized as follows:

- We formulate explicit trajectory diversity for RL-based post-training through user-specified, task-specific trajectory descriptors and set-level diversity functionals, providing an interpretable and controllable way to specify *what* should be diverse.
- We introduce Trajectory-guided Joint Policy Optimization (TJPO), a single-policy framework that converts group-level diversity into trajectory-level learning signals, enabling explicit diversity optimization within group-based RL without maintaining a population of policies.
- We validate TJPO on ALFWorld and Sokoban, showing that it improves task-specific trajectory diversity while maintaining competitive task performance. Further trajectory-level and intervention-based evaluations show that the learned diversity corresponds to distinct successful strategies and can improve compliance with user constraints and robustness to environmental disruptions without additional training.

2 PRELIMINARY

2.1 RL FOR LLMS

In RL-based post-training for LLMs, generation is modeled as a sequential decision-making process. Given an input $x \in \mathcal{X}$, the model acts as a policy that autoregressively generates a trajectory $\tau = (\tau^1, \tau^2, \ldots, \tau^T) \in \mathcal{T}$, where $\mathcal{T}$ denotes the space of all possible outputs. The policy is parameterized by θ and factorizes as

$$\pi_\theta(\tau \mid x) := \prod_{t=1}^{T} \pi_\theta(\tau^t \mid x, \tau^{<t}), \tag{1}$$

where $\tau^{<t} = (\tau^1, \ldots, \tau^{t-1})$. A reward function is then used to evaluate the generated trajectory and optimize the policy.

2.2 GROUP-BASED REINFORCEMENT LEARNING ALGORITHMS

Group Relative Policy Optimization (GRPO) GRPO eliminates the need for a separate critic model by estimating the baseline from a group of sampled trajectories. For each input x, it samples a group of trajectories $\{\tau_1, \ldots, \tau_G\}$ from the old policy π_{old}. Writing the trajectory likelihood ratio as $\rho_i(\theta) = \pi_\theta(\tau_i|x)/\pi_{\text{old}}(\tau_i|x)$, GRPO optimizes the current policy π_θ via

$$\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\Big(\rho_i(\theta)A_i, \text{clip}\big(\rho_i(\theta), 1-\epsilon, 1+\epsilon\big)A_i\Big) - \beta\mathbb{D}_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right)\right], \tag{2}$$

where the expectation is over $x \sim \mathcal{X}$ and $\{\tau_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot|x)$, ϵ and β are hyperparameters, and

$$\mathbb{D}_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(\tau_i|x)}{\pi_\theta(\tau_i|x)} - \log\frac{\pi_{\text{ref}}(\tau_i|x)}{\pi_\theta(\tau_i|x)} - 1, \quad A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{3}$$

Group-in-Group Policy Optimization (GiGPO) While GRPO is effective for single-turn generation, long-horizon agent tasks present a more challenging credit assignment problem due to sparse and delayed rewards. GiGPO extends group-based RL by introducing an additional step-level relative advantage on top of the trajectory-level advantage used in GRPO.

To provide finer-grained supervision, GiGPO constructs step-level groups using *anchor states*, i.e., repeated or aligned environment states observed across trajectories. For an action taken at step t in trajectory i, it computes a micro advantage by comparing it against other actions originating from the same anchor-state group:

$$A_{i,t}^{\text{micro}} = \frac{u_{i,t} - \text{mean}(\mathcal{H}(s_{i,t}))}{\text{std}(\mathcal{H}(s_{i,t}))}, \tag{4}$$

where $\mathcal{H}(s_{i,t})$ denotes the set of step-level utilities associated with the anchor-state group of $s_{i,t}$. The final advantage combines the trajectory-level signal from GRPO with the step-level signal:

$$\hat{A}_{i,t} = A_i + \omega A_{i,t}^{\text{micro}}, \tag{5}$$

where ω controls the strength of the micro-level credit assignment. GiGPO is particularly suitable for long-horizon LLM agent training, where useful supervision must be propagated not only across full trajectories but also across individual decision steps.

3 METHOD

We study explicit trajectory diversity in group-based policy optimization for LLMs. Our goal is to encourage *intent-aligned* behavioral variation, namely, variation along task-relevant dimensions specified by the practitioner, rather than generic randomness induced by token-level entropy or decoding noise.

3.1 PROBLEM SETUP

Following the notation in Section 2.2, for each input x we sample a group of trajectories

$$\mathcal{G}(x) = \{\tau_i\}_{i=1}^{G}, \qquad \tau_i \sim \pi_\theta(\cdot \mid x). \tag{6}$$

Each trajectory τ_i receives a task reward $r_i^{\text{task}} = R(x, \tau_i)$. Our goal is to augment this reward with an explicit diversity signal defined over the sampled trajectory group $\mathcal{G}(x)$.

3.2 INTENT-ALIGNED TRAJECTORY DESCRIPTORS

Definition 1 (Intent-Aligned Trajectory Descriptor)**.** *Given the trajectory space $\mathcal{T}$, an intent-aligned trajectory descriptor is a user-specified scalar function*

$$\phi_k : \mathcal{T} \to \mathbb{R}, \qquad k = 1, \ldots, m, \tag{7}$$

that measures one task-relevant axis of behavioral variation.

A descriptor family $\{\phi_k\}_{k=1}^{m}$ induces a trajectory-to-vector mapping

$$\mathbf{z}(\tau) = \big[\phi_1(\tau), \ldots, \phi_m(\tau)\big]^\top \in \mathbb{R}^m. \tag{8}$$

For each sampled trajectory $\tau_i \in \mathcal{G}(x)$, we write $\mathbf{z}_i = \mathbf{z}(\tau_i)$. Collecting all descriptors in the group yields the descriptor matrix

$$Z = [\mathbf{z}_1 \cdots \mathbf{z}_G]^\top \in \mathbb{R}^{G \times m}. \tag{9}$$

By construction, the descriptor family $\{\phi_k\}_{k=1}^{m}$ is intent-aligned: it explicitly specifies which forms of trajectory variation are desirable for the task, rather than relying on task-agnostic statistics over the policy's state-action distribution.

3.3 GROUP-LEVEL EXPLICIT DIVERSITY

We now define explicit diversity as a set-level function over the descriptor vectors induced by a sampled trajectory group.

Definition 2 (Explicit Diversity Functional)**.** *Given a descriptor matrix $Z \in \mathbb{R}^{G\times m}$, where each row corresponds to the descriptor vector of one sampled trajectory, an explicit diversity functional is a mapping*

$$\mathcal{D} : \mathbb{R}^{G\times m} \to \mathbb{R}, \tag{10}$$

which assigns a scalar diversity score to the trajectory group. We require $\mathcal{D}$ to be permutation-invariant with respect to the ordering of trajectories, i.e., for any $G \times G$ permutation matrix P,

$$\mathcal{D}(Z) = \mathcal{D}(PZ). \tag{11}$$

Pairwise diversity. A simple choice is the average pairwise dispersion:

$$\mathcal{D}_{\text{pair}}(Z) = \frac{2}{G(G-1)} \sum_{1\le i<j\le G} d(\mathbf{z}_i, \mathbf{z}_j), \tag{12}$$

where $d(\cdot,\cdot)$ is a distance function, e.g., Euclidean distance or cosine distance. By construction, $\mathcal{D}_{\text{pair}}$ is permutation-invariant.

Determinantal diversity. To capture set-level coverage beyond pairwise comparisons, we also consider a determinantal form. Let

$$K_{ij} = k(\mathbf{z}_i, \mathbf{z}_j), \tag{13}$$

where $k(\cdot,\cdot)$ is a positive semidefinite kernel, such as an RBF kernel. We define

$$\mathcal{D}_{\text{det}}(Z) = \log\det(K + \eta I), \tag{14}$$

where $\eta > 0$ ensures numerical stability. Maximizing (14) encourages the sampled trajectories to span a larger volume in the descriptor space, thereby reducing redundancy. This functional is also permutation-invariant, since permuting the rows of Z only induces a simultaneous row-column permutation of K, which leaves the determinant unchanged.

The formulation above is intentionally modular: any permutation-invariant functional over the descriptor set can be plugged into our framework.

3.4 FROM GROUP DIVERSITY TO TRAJECTORY-LEVEL CREDIT

A subtle but important issue arises when combining explicit diversity with GRPO-style training. Since GRPO computes *group-relative* advantages, adding the same group-level scalar bonus to every trajectory has no effect after baseline subtraction. Indeed, the shared term in

$$\bar{r}_i = r_i^{\text{task}} + \lambda\mathcal{D}(Z) \qquad \forall i, \tag{15}$$

vanishes after group centering, so diversity must be converted into *trajectory-level* credit.

For pairwise diversity, we use the average distance to the remaining group members:

$$c_i^{\text{pair}}(Z) = \frac{1}{G-1} \sum_{j\neq i} d(\mathbf{z}_i, \mathbf{z}_j). \tag{16}$$

As derived in Appendix A, this credit is a positive affine transformation of the exact leave-one-out change within a fixed group and therefore yields identical group-standardized values.

For determinantal diversity, with $A = K + \eta I$, we use

$$c_i^{\text{det}}(Z) = \frac{\det(A)}{\det(A_{-i,-i})} = \exp(\mathcal{D}_{\text{det}}(Z) - \mathcal{D}_{\text{det}}(Z_{-i}))\,. \tag{17}$$

This is the exponential of the leave-one-out log-determinant change.

Finally, we standardize the credits within each sampled group and add them to the GiGPO advantage:

$$\hat{c}_i = \frac{c_i - \bar{c}}{\max\{\mathrm{std}(c), \epsilon\}}, \tag{18}$$

$$\widetilde{A}_{i,t} = \hat{A}_{i,t} + \lambda \hat{c}_i, \tag{19}$$

where $\epsilon > 0$ handles near-zero variance and $\lambda \geq 0$ controls the task–diversity trade-off. The trajectory-level credit is shared across the valid response tokens associated with trajectory i.

Our method preserves the simplicity of GRPO-style training while explicitly optimizing task-relevant trajectory diversity. The key difference from implicit diversity regularizers is that diversity is defined over full-trajectory descriptors chosen to match the intended behavioral variation of each environment, rather than generic statistics of token-level stochasticity.

Taken together, our framework cleanly separates *what* should be diverse from *how* diversity is optimized. The descriptor family $\{\phi_k\}$ specifies the intended axes of variation, while the diversity functional $\mathcal{D}$ specifies how a group of sampled trajectories should spread out in the resulting descriptor space. This modular design allows the same optimization framework to be instantiated across diverse environments with task-specific notions of meaningful diversity. We term our method *Trajectory-guided Joint Policy Optimization* (TJPO).

4 Experiments

We evaluate TJPO through the following questions:

Q1. Can TJPO improve task-specific diversity while maintaining strong task performance?

Q2. How do trajectory descriptors and diversity functionals shape the learned behavioral variation?

Q3. How does the diversity coefficient λ affect the diversity–performance trade-off?

Q4. Does the resulting behavioral diversity provide practical benefits under user constraints and environmental disruptions?

4.1 Experimental Setup

Environments. We evaluate on ALFWorld (Shridhar et al., 2021) and Sokoban (Schrader, 2018). ALFWorld requires an agent to complete household manipulation tasks through multi-step text interactions, where trajectories consist of observations, reasoning, and textual actions. Sokoban is a grid-based box-pushing puzzle in which the agent must plan action sequences to push boxes onto target cells. We provide detailed environment descriptions in Appendix C.1.

Models. We use Qwen2.5-7B-Instruct as the base model for ALFWorld and Qwen3-VL-4B-Instruct for Sokoban. Within each environment, all trained policies start from the same base model and use the same rollout budget, optimization hyperparameters, and evaluation protocol. Our decoding-only baseline reuses the trained GiGPO checkpoint without any additional optimization.

Training backbone. We implement our method on top of GiGPO (Feng et al., 2025b), a group-based RL algorithm for long-horizon LLM agent training. For each input instance, the policy samples a group of trajectories. We compute a trajectory-level diversity credit for each trajectory, standardize the credits within the sampled group, and add the resulting signal to the GiGPO advantage as described in Equation 19. Implementation details are provided in Appendix C.2.

Evaluation metrics. We report both task performance and explicit diversity. For task performance, we use success rate and average trajectory reward. For diversity, we sample N_{eval} trajectories per input and compute descriptor-based diversity over the successful subset of sampled trajectories using the corresponding evaluation descriptor.

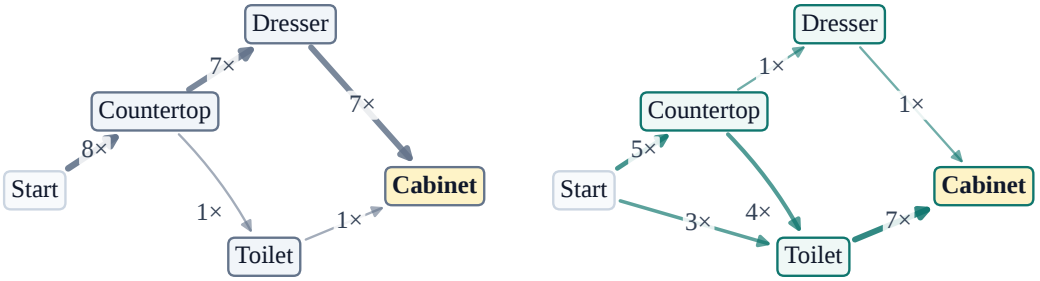
Baselines. Our primary baseline is GiGPO without explicit diversity optimization. For ALFWorld, we also evaluate GiGPO+PP and GiGPO+RP, inference-only baselines that apply a presence penalty of 0.5 or repetition penalty of 1.05 to the same checkpoint while leaving all other evaluation settings

Table 1: Pairwise diversity results on ALFWorld.

Method	Success ↑	$D_{\text{act}}^{\text{pair}}$ ↑	$D_{\text{loc}}^{\text{pair}}$ ↑
GiGPO	0.9160	0.0495	0.0858
GiGPO + PP	0.9082	0.0458	0.0895
GiGPO + RP	0.9229	0.0494	0.0873
$\text{TJPO}_{\text{act}}^{\text{pair}}$	0.8682	**0.0780**	0.1002
$\text{TJPO}_{\text{loc}}^{\text{pair}}$	0.9053	0.0601	**0.1316**

Table 2: Determinantal diversity results on ALFWorld.

Method	Success ↑	$D_{\text{act}}^{\text{dpp}}$ ↑ ($\times 10^{-3}$)	$D_{\text{loc}}^{\text{dpp}}$ ↑ ($\times 10^{-3}$)
GiGPO	0.9160	1.887	2.500
GiGPO + PP	0.9082	1.858	2.063
GiGPO + RP	0.9229	1.135	1.544
$\text{TJPO}_{\text{act}}^{\text{dpp}}$	0.9160	**4.100**	6.655
$\text{TJPO}_{\text{loc}}^{\text{dpp}}$	0.8779	3.058	**7.940**



(a) GiGPO navigation strategies.

(b) TJPO navigation strategies.

Figure 2: Successful navigation strategies for "put a spraybottle in cabinet". TJPO uses the location descriptor with pairwise diversity and $\lambda = 0.3$. Both methods succeed in all eight rollouts. Nodes denote visited locations; edge labels and widths encode navigation transition counts across the eight rollouts.

unchanged. We compare action-type and location descriptors with pairwise or determinantal diversity. For Sokoban, we compare GiGPO with two task-specific descriptors, denoted as direction-transition and spatial-coverage, which are detailed in Appendix D.

4.2 Main Results on ALFWorld

Tables 1 and 2 report the pairwise and determinantal diversity results on ALFWorld, respectively. All TJPO variants use $\lambda = 0.3$. For determinantal diversity D^{dpp}, we use the determinant-root score defined in Appendix E, with values reported in units of 10^{-3}. We compare standard GiGPO, two decoding-only penalty baselines, and four TJPO variants obtained by combining two task-specific trajectory descriptors, action type and location, with either pairwise or determinantal diversity. The results show that TJPO improves explicit trajectory diversity under the intended descriptor metrics while maintaining competitive task performance. For the action-type descriptor, TJPO with pairwise diversity increases $D_{\text{act}}^{\text{pair}}$ from 0.0495 to 0.0780, with a success rate of 0.8682; the determinantal variant increases $D_{\text{act}}^{\text{dpp}}$ from 1.887×10^{-3} to 4.100×10^{-3} with a success rate of 0.9160. For the location descriptor, TJPO with pairwise diversity increases $D_{\text{loc}}^{\text{pair}}$ from 0.0858 to 0.1316, with a success rate of 0.9053; the determinantal variant increases $D_{\text{loc}}^{\text{dpp}}$ from 2.500×10^{-3} to 7.940×10^{-3}, with a success rate of 0.8779. These results indicate that user-specified descriptors provide a controllable interface for shaping which aspects of trajectory behavior become diverse. The cross-metric results show that the descriptor determines which aspects of behavior become more diverse. Action-type optimization primarily changes the composition of action categories, whereas location-level optimization encourages broader variation in location visitation patterns.

Table 3: Main results on Sokoban.

Method	Success ↑	$D_{\text{dir}}^{\text{pair}}$ ↑	$D_{\text{spa}}^{\text{pair}}$ ↑
GiGPO	0.8281	0.0146	0.0092
TJPO$_{\text{dir}}^{\text{pair}}$	0.8428	**0.0675**	0.0503
TJPO$_{\text{spc}}^{\text{pair}}$	0.8057	0.0595	**0.0518**

Table 4: Ablation on the λ on ALFWorld.

Method	λ	Success ↑	$D_{\text{loc}}^{\text{pair}}$ ↑	Avg. Steps ↓
TJPO	0	0.9160	0.0858	9.2217
TJPO	0.1	0.9209	0.0942	9.5811
TJPO	0.3	0.9053	0.1316	9.6947
TJPO	0.5	0.9004	0.1356	8.1150

Table 5: Practical benefits of behavioral diversity on ALFWorld.

Method	Standard Success ↑	User Constraints Success ↑	Disruption Success ↑
GiGPO	**0.9917**	0.7708	0.7792
TJPO	0.9750	**0.8542**	**0.8542**

4.3 Main Results on Sokoban

Table 3 reports the main results on Sokoban. All TJPO variants use $\lambda = 0.3$ and pairwise diversity. Compared with the GiGPO baseline, TJPO improves explicit trajectory diversity under both task-specific descriptors. When optimizing the direction-transition descriptor, TJPO increases $D_{\text{dir}}^{\text{pair}}$ from 0.0146 to 0.0675, with a success rate of 0.8428. When optimizing the spatial-coverage descriptor, TJPO increases $D_{\text{spa}}^{\text{pair}}$ from 0.0092 to 0.0518, with a success rate of 0.8057. The results show that TJPO can substantially improve planning-level trajectory diversity while maintaining competitive task performance. Together with the ALFWorld results in Section 4.2, these findings suggest that our framework is not limited to tool-use trajectories, but also applies to planning-intensive game environments where meaningful diversity is reflected in movement patterns and state-space coverage.

4.4 Case Study

Figure 2 compares the successful navigation strategies of GiGPO and TJPO for the task "put a spraybottle in cabinet". GiGPO follows the same route in seven of eight rollouts. TJPO instead distributes its rollouts across three successful navigation strategies, including a more direct route through the Toilet. This example illustrates that explicit location descriptor encourages alternative efficient solution modes rather than merely increasing a scalar diversity score. We further expand the route-level case study in Appendix F.

4.5 Ablation on Diversity Weight

Table 4 studies the effect of the diversity coefficient λ. For the location descriptor, increasing λ improves $D_{\text{loc}}^{\text{pair}}$, from 0.0858 at $\lambda = 0$ to 0.1356 at $\lambda = 0.5$, while maintaining competitive task performance. The mean successful-trajectory length remains comparable at $\lambda \leq 0.3$ and decreases to 8.1150 steps at $\lambda = 0.5$, indicating that the diversity gain does not result from longer successful interactions.

4.6 Practical Benefits of Behavioral Diversity

Beyond diversity scores, we examine whether behavioral diversity helps agents satisfy user-specific requirements and cope with environmental disruptions. We use location access in ALFWorld as a concrete example: *User Constraints* specifies locations the agent must avoid, whereas *Disruption* makes task-relevant locations unavailable. *Standard* denotes the original task without either intervention. We compare GiGPO with location-pairwise TJPO at $\lambda = 0.5$ on tasks with verified alternative solutions. Details are provided in Appendix G.

Table 5 shows that TJPO improves compliant success under user constraints from 77.08% to 85.42% and task success under environmental disruption from 77.92% to 85.42%. These results illustrate the practical value of behavioral diversity in this setting: producing successful behaviors that meet user requirements and maintaining task completion when the environment is disrupted.

5 Related work

Diversity in RL Prior work on policy diversity in deep reinforcement learning can be broadly organized according to the role that diversity plays in the learning process (Wu et al., 2023). One line of work leverages diversity to facilitate exploration, where diverse behaviors are encouraged mainly as an intermediate mechanism for improving final task performance (Hong et al., 2018; Eysenbach et al., 2018; Parker-Holder et al., 2020; Conti et al., 2018; Peng et al., 2020; Yao et al., 2023). A second line incorporates diversity through constrained optimization, either by maximizing task quality under diversity constraints or by maximizing diversity while enforcing quality constraints (Masood & Doshi-Velez, 2019; Zhang et al., 2019; Ghasemi et al., 2021; Zahavy et al., 2021; Zhou et al., 2022). A third line jointly optimizes quality and diversity, commonly referred to as Quality-Diversity RL (Cideron et al., 2020; Pierrot et al., 2022; Tjanaka et al., 2022; Batra et al., 2023; Yang et al., 2025). Our work extends the second paradigm to RL for LLM training.

Diversity in LLMs Prior work has explored diversity in LLMs from several perspectives, including supervised fine-tuning, preference optimization, RL-based post-training, diversity evaluation. In supervised and preference-based post-training, existing methods aim to preserve or improve response diversity during alignment, often by modifying the training objective, constructing diverse preference pairs, or encouraging diverse high-quality generations (Li et al., 2025c; Lanchantin et al., 2025; Chung et al., 2025; Chen et al., 2025b). Complementary studies evaluate linguistic, semantic, and epistemic diversity in LLM outputs, showing that diversity is an important dimension beyond average quality or accuracy (Shypula et al., 2025; Guo et al., 2025b; Wright et al., 2025). In RL-related settings, recent work has begun to study diversity in LLM reasoning, while other approaches use divergence choices or entropy-based objectives to indirectly preserve exploration and mitigate diversity collapse (Yao et al., 2025; Li et al., 2025a). These works are closely related to ours, but they typically define or encourage diversity through variation among token-level diversity, embedding-based semantic diversity, or implicit distributional regularization. In contrast, our work focuses on *explicit trajectory diversity*: diversity is defined through user-specified, task-specific trajectory descriptors that directly encode the intended behavioral variations of each environment, such as reasoning structures, planning patterns, or tool-use trajectories, and is optimized within RL-based post-training.

6 Conclusion, Limitations, and Discussion

We study explicit trajectory diversity in RL-based post-training for LLMs. Unlike implicit notions of diversity that arise from sampling randomness, entropy regularization, or distributional constraints, our framework defines diversity through user-specified, task-specific trajectory descriptors. These descriptors make explicit which behavioral dimensions should vary for a given environment, such as planning patterns and tool-use trajectories. Building on this formulation, we introduce a framework to optimize diversity over sampled trajectory groups. By converting set-level diversity into trajectory-level learning signals, the proposed method can be integrated into group-based RL training, such as GiGPO. Across Sokoban and agentic tasks, our experiments show that explicit trajectory diversity can be consistently improved while maintaining strong task performance.

While our results are encouraging, several aspects merit further study. First, our framework relies on task-specific trajectory descriptors. This design is intentional, since explicit diversity should reflect the behavioral variations that matter for the target environment. At the same time, different descriptor choices may emphasize different aspects of behavior, and developing better interfaces or reusable descriptor templates across task families, such as reasoning, planning, tool use, and embodied interaction, would make the framework easier to apply while preserving its user-controlled nature. Second, although our experiments span planning and agentic tasks, broader evaluation on richer agent settings would further clarify the generality of explicit trajectory diversity. Promising directions include larger-scale tool ecosystems, open-ended web environments, multi-agent settings, longer horizons, larger action spaces, and more complex environmental shifts. Overall, our results suggest that explicitly defining and optimizing task-aligned trajectory diversity is a useful step toward more controllable, robust, and adaptable LLM agents.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the paper and preliminary coding. We have not used generative AI tools for generating the research idea and designing the algorithm. We have reviewed all AI-assisted work. For example, the polished text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We confirm that all authors of this submission have read and agree to abide by the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Our experiments are reproducible. We provide the complete code in the supplementary material to facilitate reproduction of all reported results.

REFERENCES

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495, 2024.

Sumeet Batra, Bryon Tjanaka, Matthew C Fontaine, Aleksei Petrenko, Stefanos Nikolaidis, and Gaurav Sukhatme. Proximal policy gradient arborescence for quality diversity reinforcement learning. *arXiv preprint arXiv:2305.13795*, 2023.

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

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Yannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. *arXiv preprint arXiv:2509.04784*, 2025b.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying large language model post-training for diverse creative writing. *arXiv preprint arXiv:2503.17126*, 2025.

Geoffrey Cideron, Thomas Pierrot, Nicolas Perrin, Karim Beguir, and Olivier Sigaud. Qd-rl: Efficient mixing of quality and diversity in reinforcement learning. corr abs/2006.08505 (2020). *arXiv preprint arXiv:2006.08505*, 2020.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *Advances in neural information processing systems*, 31, 2018.

Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.

Shangheng Du, Jiabao Zhao, Jinxin Shi, Zhentao Xie, Xin Jiang, Yanhong Bai, and Liang He. A survey on the optimization of large language model-based agents. *ACM Comput. Surv.*, 58(9):223:1–223:37, 2026. doi: 10.1145/3789261. URL https://doi.org/10.1145/3789261.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Lang Feng, Weihao Tan, Zhiyi Lyu, Longtao Zheng, Haiyang Xu, Ming Yan, Fei Huang, and Bo An. Towards efficient online tuning of vlm agents via counterfactual soft reinforcement learning. *arXiv preprint arXiv:2505.03792*, 2025a.

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

Mahsa Ghasemi, Evan Scope Crafts, Bo Zhao, and Ufuk Topcu. Multiple plans are better than one: Diverse stochastic planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pp. 140–148, 2021.

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*, 2025a.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. Benchmarking linguistic diversity of large language models. *Transactions of the Association for Computational Linguistics*, 13:1507–1526, 2025b.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Ameya Prabhu, and Matthias Bethge. Curatedthoughts: Data curation for rl training datasets, 2025. URL https://huggingface.co/datasets/bethgelab/CuratedThoughts.

Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Jack Lanchantin, Angelica Chen, Shehzaad Dhuliawala, Ping Yu, Jason Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Diverse preference optimization. *arXiv preprint arXiv:2501.18101*, 2025.

Long Li, Zhijian Zhou, Jiaran Hao, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, et al. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.07430*, 2025a.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025b.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025c.

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

Muhammad A Masood and Finale Doshi-Velez. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *arXiv preprint arXiv:1906.00088*, 2019.

Chunyang Meng, Shijie Song, Haogang Tong, Maolin Pan, and Yang Yu. Deepscaler: Holistic autoscaling for microservices based on spatiotemporal gnn with adaptive graph learning. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 53–65. IEEE, 2023.

OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms`, 2024.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18050–18062, 2020.

Zhenghao Peng, Hao Sun, and Bolei Zhou. Non-local policy optimization via diversity-regularized collaborative exploration. *arXiv preprint arXiv:2006.07781*, 2020.

Thomas Pierrot, Valentin Macé, Felix Chalumeau, Arthur Flajolet, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin-Gilbert. Diversity policy gradient for sample efficient quality-diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1075–1083, 2022.

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

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

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. URL `https://arxiv.org/abs/2010.03768`.

Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Evaluating the diversity and quality of llm generated content. *arXiv preprint arXiv:2504.12522*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Bryon Tjanaka, Matthew C Fontaine, Julian Togelius, and Stefanos Nikolaidis. Approximating gradients for differentiable quality diversity in reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1102–1111, 2022.

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

Dustin Wright, Sarah Masud, Jared Moore, Srishti Yadav, Maria Antoniak, Peter Ebert Christensen, Chan Young Park, and Isabelle Augenstein. Epistemic diversity and knowledge collapse in large language models. *arXiv preprint arXiv:2510.04226*, 2025.

Shuang Wu, Jian Yao, Haobo Fu, Ye Tian, Chao Qian, Yaodong Yang, Qiang Fu, and Yang Wei. Quality-similar diversity via population based reinforcement learning. In *The eleventh international conference on learning representations*, 2023.

Hanlin Yang, Jian Yao, Weiming Liu, Qing Wang, Hanmin Qin, Kirk Tang, Jiechao Xiong, Chao Yu, Kai Li, Junliang Xing, et al. Diverse policies recovering via pointwise mutual information weighted imitation learning. In *International Conference on Learning Representations*, volume 2025, pp. 24860–24878, 2025.

Jian Yao, Weiming Liu, Haobo Fu, Yaodong Yang, Stephen McAleer, Qiang Fu, and Wei Yang. Policy space diversity for non-transitive games. *Advances in Neural Information Processing Systems*, 36:67771–67793, 2023.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 2025.

Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025a.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025b.

Tom Zahavy, Brendan O'Donoghue, Andre Barreto, Volodymyr Mnih, Sebastian Flennerhag, and Satinder Singh. Discovering diverse nearly optimal policies with successor features. *arXiv preprint arXiv:2106.00669*, 2021.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. `https://hkust-nlp.notion.site/simplerl-reason`, 2025. Notion Blog.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The landscape of agentic reinforcement learning for llms: A survey. *Trans. Mach. Learn. Res.*, 2026, 2026. URL `https://openreview.net/forum?id=RY19y2RI1O`.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, et al. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025.

Yunbo Zhang, Wenhao Yu, and Greg Turk. Learning novel policies for tasks. In *International Conference on Machine Learning*, pp. 7483–7492. PMLR, 2019.

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

Zihan Zhou, Wei Fu, Bingliang Zhang, and Yi Wu. Continuously discovering novel strategies via reward-switching policy optimization. *arXiv preprint arXiv:2204.02246*, 2022.

A Derivation and Implementation of Trajectory-Level Diversity Credits

Pairwise diversity credit. For a sampled group of G trajectory descriptors, define

$$\mathcal{D}_{\text{pair}}(Z) = \frac{2}{G(G-1)} \sum_{1 \le a < b \le G} d(\mathbf{z}_a, \mathbf{z}_b) \tag{20}$$

and

$$c_i^{\text{pair}}(Z) = \frac{1}{G-1} \sum_{j \neq i} d(\mathbf{z}_i, \mathbf{z}_j). \tag{21}$$

Let $S = \sum_{a<b} d(\mathbf{z}_a, \mathbf{z}_b)$ and $s_i = \sum_{j \neq i} d(\mathbf{z}_i, \mathbf{z}_j)$. Since

$$\mathcal{D}_{\text{pair}}(Z_{-i}) = \frac{2(S - s_i)}{(G-1)(G-2)}, \tag{22}$$

the exact leave-one-out change satisfies

$$\mathcal{D}_{\text{pair}}(Z) - \mathcal{D}_{\text{pair}}(Z_{-i}) = \frac{2}{G-2} \left[c_i^{\text{pair}}(Z) - \mathcal{D}_{\text{pair}}(Z) \right]. \tag{23}$$

Moreover, every unordered pair contributes to two credits, so

$$\frac{1}{G} \sum_{i=1}^{G} c_i^{\text{pair}}(Z) = \mathcal{D}_{\text{pair}}(Z). \tag{24}$$

Thus, within a fixed group, the exact leave-one-out change is a positive affine transformation of c_i^{pair}. The two therefore produce identical values after group-wise standardization.

Determinantal diversity credit. We construct

$$K_{ij} = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{\sigma} \right) \tag{25}$$

and let $A = K + \eta I$. The implemented credit is

$$c_i^{\text{det}}(Z) = \frac{\det(A)}{\det(A_{-i,-i})}. \tag{26}$$

Equivalently,

$$\log c_i^{\text{det}} = \log\det(A) - \log\det(A_{-i,-i}), \tag{27}$$

so the determinant ratio is the exponential of the leave-one-out log-determinant change. By the Schur-complement identity,

$$c_i^{\text{det}}(Z) = A_{ii} - A_{i,-i} A_{-i,-i}^{-1} A_{-i,i}. \tag{28}$$

Numerical implementation. The pairwise credit uses Euclidean distance. For the determinantal credit, we use $\sigma = 10$ and diagonal jitter $\eta = 10^{-4}$. Raw determinant-ratio credits are divided by their within-group maximum before standardization; this common positive scaling does not affect the subsequent group Z-score. If the regularized full determinant is at most 10^{-10}, we use the mean off-diagonal kernel dissimilarity $1 - K_{ij}$ as a numerically robust fallback. If an individual leave-one-out determinant is at most 10^{-10} while the full determinant is valid, its raw credit is set to one. Groups with zero credit variance receive zero normalized diversity advantage.

B More Discussion on Related Work

Given the recent surge of interest in RL for LLMs and agents, there is a substantial body of relevant work worth discussing. We provide a more detailed discussion of these prior studies here.

RL for LLMs and Agents. Recent LLMs have shown rapid progress in complex reasoning, with strong performance demonstrated by frontier reasoning models such as OpenAI-o1 (OpenAI, 2024), DeepSeek-R1 (Guo et al., 2025a), and Kimi-k1.5 (Team et al., 2025). A representative line of work follows the R1-zero training paradigm introduced by DeepSeek-R1 (Guo et al., 2025a), which combines group-based policy optimization with rule-based rewards for final-answer correctness and formatting. Subsequent studies have improved this paradigm along two main directions: improving data quality, curriculum construction, and rollout filtering (Li et al., 2025b; Meng et al., 2023; He et al., 2025; Yu et al., 2025; Hu et al., 2025; Hochlehnert et al., 2025), and refining RL algorithms for more stable and efficient post-training. The latter includes PPO-style and value-based methods for improved advantage estimation and long-chain credit assignment (Zeng et al., 2025; Kazemnejad et al., 2024; Yuan et al., 2025b;a), GRPO-style methods that address instability, biased objectives, length or difficulty biases, and sample efficiency (Yu et al., 2025; Liu et al., 2025; Zhang et al., 2025), as well as REINFORCE-like methods that improve stability or scalability (Team et al., 2025; Hu, 2025; Chu et al., 2025).

Beyond static reasoning tasks, RL has also been increasingly used to train LLM agents that interact with external environments over multiple turns, including web navigation, application control, embodied tasks, and other long-horizon interactive settings. Some methods improve agent learning through hierarchical training or guided search, often targeting web-based tasks such as WebShop (Zhou et al., 2024; Putta et al., 2024). Another line studies online RL for interactive agents in device-control or vision-language environments, where exploration in large action spaces is a central challenge (Bai et al., 2024; Feng et al., 2025a). More recent work adapts group-based or trajectory-level RL to long-horizon agent training, focusing on full-trajectory optimization, scalable multi-turn learning, or finer-grained credit assignment (Chen et al., 2025a; Wang et al., 2025; Feng et al., 2025b). These works primarily target training stability, exploration efficiency, and credit assignment in interactive environments.

C Additional Experimental Details

C.1 Environment Details

ALFWorld. ALFWorld (Shridhar et al., 2021) is a text-based interactive environment aligned with embodied household tasks. Each episode provides a natural-language goal and textual observations of the current environment. At each step, the agent outputs a textual action, such as navigation, object manipulation, or environment inspection. An episode is considered successful if the agent completes the specified household task within the step budget. In our experiments, a trajectory consists of the full sequence of observations, reasoning tokens, actions, and environment feedback:

$$\tau = (o_1, a_1, o_2, a_2, \ldots, o_T, a_T). \tag{29}$$

The task reward is defined as

$$r^{\text{task}} = \mathbf{1}[\text{task success}]. \tag{30}$$

Sokoban. Sokoban (Schrader, 2018) is a grid-based box-pushing puzzle game. The agent moves on a grid and must push boxes onto target cells. The agent can push boxes but cannot pull them, making the environment sensitive to irreversible action sequences and long-horizon planning. An episode is successful (and receive positive reward) if all boxes are placed on target cells within the step budget. A trajectory consists of the sequence of grid states and actions:

$$\tau = (s_1, a_1, s_2, a_2, \ldots, s_T, a_T). \tag{31}$$

We use Sokoban to evaluate whether explicit diversity can capture planning-level behavioral variation rather than tool-use variation.

C.2 Training Implementation

We implement our method on top of GiGPO (Feng et al., 2025b) using the verl training framework (Sheng et al., 2025). For each input instance, we sample a group of G trajectories from the current policy. We compute the task reward r_i^{task} and descriptor vector $\mathbf{z}_i$ for each trajectory. Given

Table 6: Environment-specific experimental setup.

Setting	ALFWorld	Sokoban
Base model	Qwen2.5-7B-Instruct	Qwen3-VL-4B-Instruct
Input modality	Text	Vision-language
Environment	AlfredTWEnv	Sokoban
Environment observation	Text	RGB image
Max environment steps	50	15
Training data size	16	32
Total epochs	150	200
Advantage normalization mode	mean_std_norm	mean_norm
Max prompt length	2048	1024
PPO mini-batch size	256	64
PPO micro-batch size per GPU	16	8
Validation frequency (steps)	5	10

the descriptor matrix Z, we compute the diversity credit $c_i(Z)$ and standardize it within trajectories sampled for the same input:

$$\hat{c}_i = \frac{c_i - \bar{c}}{\max\{\mathrm{std}(c), 10^{-6}\}}. \tag{32}$$

For both ALFWorld and Sokoban, the standardized credit is added to the GiGPO advantage:

$$\widetilde{A}_{i,t} = A^{\mathrm{GiGPO}}_{i,t} + \lambda \hat{c}_i. \tag{33}$$

The trajectory-level signal is broadcast over the valid response tokens associated with trajectory i. The environment-level setup is summarized in Table 6, and the shared optimization and evaluation settings are reported in Table 7.

C.3 Reproducibility

Runs and randomness. Each reported model configuration is obtained from one training run. The training scripts set the environment seed to 0 for both ALFWorld and Sokoban. Evaluation uses stochastic decoding with temperature 0.4. For the main ALFWorld comparison, every checkpoint is evaluated on the same 128 prompts with eight rollouts per prompt, yielding 1,024 trajectories per checkpoint. The GiGPO+PP and GiGPO+RP baselines reuse the selected GiGPO checkpoint and change only the vLLM presence penalty or repetition penalty, respectively. They receive no additional training and are not used for checkpoint selection. Diversity metrics are computed within prompt groups over successful trajectories.

Compute and software. All reported training runs use one node with eight GPUs and tensor parallel size one. The experiments use eight NVIDIA H20 GPUs with 144 GB memory per GPU. Training is implemented in the PyTorch-based verl framework, with Ray for environment workers and vLLM for rollout generation; the environment uses vLLM 0.17, and pins Transformers 4.51.1.

D Trajectory Descriptor Details

This section describes the task-specific trajectory descriptors used in our experiments. Each descriptor maps a trajectory τ to a vector $\mathbf{z}(\tau) \in \mathbb{R}^m$. All descriptor values are computed after trajectory generation and are used as policy-optimization signals during RL training.

D.1 ALFWorld Descriptors

Action-type descriptor. The action-type descriptor captures the distribution of high-level action categories used by the agent across a trajectory. We parse each environment action into one of m=15 canonical ALFWorld action types: navigation (*go to*), container manipulation (*open*, *close*), object manipulation (*pick up*, *put*, *take*, *slice*), state-changing operations (*clean*, *heat*, *cool*, *toggle*,

Table 7: Shared training and evaluation hyperparameters.

Hyperparameter	**Value**
Training framework	verl
RL backbone	GiGPO
Group size G	8
Max response length	512
Learning rate	1×10^{-6}
KL loss type	low-variance KL
KL loss coefficient	0.01
Discount factor γ	0.95
GiGPO step-advantage weight ω	1.0
Diversity integration mode	advantage
Diversity-credit normalization	group mean/std
Pairwise distance	Euclidean
DPP kernel bandwidth σ	10
DPP diagonal jitter η	10^{-4}
Diversity-credit standard-deviation floor	10^{-6}
Invalid action penalty coefficient	0.1
Validation temperature	0.4
Standard evaluation presence penalty	0.0
Frequency penalty	0.0
Standard repetition penalty	1.0
Stochastic validation decoding	yes
Evaluation trajectories per input N_{eval}	8
Number of GPUs	8
Number of nodes	1
Tensor parallel size	1
GPU memory utilization	0.6

use), and inspection (*examine*, *look*, *inventory*). Action-type matching is performed by string-prefix lookup on the parsed action text. Let $\mathcal{A}_{\text{type}} = \{c_1, \ldots, c_m\}$ denote the action-category set. For trajectory $\tau = \{a_1, \ldots, a_T\}$, the descriptor is the normalized count vector

$$\phi_k^{\text{action}}(\tau) \;=\; \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[\text{type}(a_t) = c_k]\,, \tag{34}$$

for $k = 1, \ldots, m$, yielding $\phi^{\text{action}}(\tau) \in \Delta^{m-1} \subset \mathbb{R}^{15}$. This descriptor encourages diversity in how the agent structures its interaction process, e.g., whether it relies more on navigation, inspection, or manipulation, and is invariant to the specific objects or locations involved.

Location descriptor. The location descriptor captures the distribution of navigation destinations visited by the agent during a trajectory. For an ALFWorld input x, let $\mathcal{L}(x) = \{\ell_1, \ldots, \ell_n\}$ denote the locations available in the corresponding environment. We parse each navigation action of the form *go to* ℓ_k and define the normalized location-frequency vector

$$\phi_k^{\text{location}}(\tau) \;=\; \frac{1}{Z(\tau)} \sum_{t=1}^{T} \mathbf{1}\left[\begin{smallmatrix}\text{type}(a_t)=\textit{go to}\\ \text{location}(a_t)=\ell_k\end{smallmatrix}\right], \tag{35}$$

for $k = 1, \ldots, n$, where $Z(\tau)$ is the total number of parsed navigation actions. This yields $\phi^{\text{location}}(\tau) \in \Delta^{n-1} \subset \mathbb{R}^{|\mathcal{L}(x)|}$. The descriptor encourages diversity in route choice and location visitation patterns while remaining invariant to non-navigation action composition.

D.2 Sokoban Descriptors

For Sokoban we instantiate two complementary trajectory-level behavior descriptors that operate on the agent's emitted move sequence: a *direction-transition* descriptor that summarizes transitions

Table 8: Summary of trajectory descriptors used in our experiments.

Environment	Descriptor	What it captures	Vector type
ALFWorld	action_type	High-level action	normalized histogram ($\mathbb{R}^{15}$)
ALFWorld	location	Navigation destination usage	normalized histogram ($\mathbb{R}^{\lvert\mathcal{L}(x)\rvert}$)
Sokoban	direction_transition	Consecutive direction pairs	normalized histogram ($\mathbb{R}^{16}$)
Sokoban	spatial_coverage	Visited grid-cell distribution	normalized 2D occupancy ($\mathbb{R}^{7\times7}$)

between consecutive move directions, and a higher-dimensional *spatial-coverage* descriptor that captures *where* on the grid the agent traverses. Both descriptors are extracted from the action stream by parsing action tags inside the agent's response and are ℓ_1-normalized to lie on the probability simplex.

Direction-transition descriptor. The direction-transition descriptor captures the distribution of consecutive low-level move-direction pairs used by the agent. Let $\mathcal{D} = \{d_1, d_2, d_3, d_4\} = \{\textit{up}, \textit{down}, \textit{left}, \textit{right}\}$ denote the four cardinal moves, and let T be the number of valid move actions in trajectory τ. For each ordered pair $(d_p, d_q) \in \mathcal{D} \times \mathcal{D}$, we define

$$\phi^{\mathrm{dir}}_{p,q}(\tau) \;=\; \frac{1}{T-1}\sum_{t=1}^{T-1} \mathbf{1}\left[\begin{smallmatrix}\mathrm{dir}(a_t)=d_p\\ \mathrm{dir}(a_{t+1})=d_q\end{smallmatrix}\right], \tag{36}$$

yielding a normalized transition histogram $\phi^{\mathrm{dir}}(\tau) \in \Delta^{15} \subset \mathbb{R}^{16}$. This descriptor is invariant to the absolute position on the board and encourages diversity in movement patterns, including straight runs, reversals, and changes between horizontal and vertical motion.

Spatial-coverage descriptor. The spatial-coverage descriptor captures *where* the agent moves on the grid by simulating its trajectory in a board-local coordinate frame. We instantiate a $H\times W$ relative grid (we use $H{=}W{=}7$ in our experiments) centered at the agent's start cell, and roll out the action sequence with a clipping policy that keeps the simulated agent inside the grid. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ with $m{=}HW$ enumerate the grid cells, and let $c_t \in \mathcal{G}$ be the simulated cell at step t (with c_0 the start cell). The descriptor is the normalized visitation count

$$\phi^{\mathrm{cov}}_k(\tau) \;=\; \frac{1}{Z(\tau)}\sum_{t=0}^{T} \mathbf{1}[c_t = g_k], \qquad k = 1, \ldots, m, \tag{37}$$

where $Z(\tau) = T + 1$ normalizes the histogram to the simplex, $\phi^{\mathrm{cov}}(\tau) \in \Delta^{m-1} \subset \mathbb{R}^{49}$. Because the simulator anchors the origin at the agent's start cell rather than the absolute board frame, two trajectories on the same puzzle that visit the same physical cells yield identical descriptors, while trajectories taking distinct routes are pulled apart in ϕ^{cov}-space. This descriptor encourages diversity in spatial exploration and route choice, and is complementary to ϕ^{dir}: two policies can share a similar direction distribution yet differ sharply in which grid cells they cover.

D.3 Descriptor Summary

Table 8 summarizes how the four descriptors capture complementary behavioral variation in action composition, location visitation, direction transitions, and spatial coverage.

E Determinantal Diversity Evaluation metric.

For the determinantal comparison, we report a size-normalized evaluation score derived from the log-determinant functional. Let Z_{succ} contain the descriptors of the n successful trajectories in an evaluation group, and let K_{succ} be their RBF Gram matrix. For $n \geq 2$, we define

$$D^{\mathrm{dpp}}(Z_{\mathrm{succ}}) = \exp\left(\frac{1}{n}\log\det(K_{\mathrm{succ}} + \eta I)\right) = \det(K_{\mathrm{succ}} + \eta I)^{1/n}, \tag{38}$$

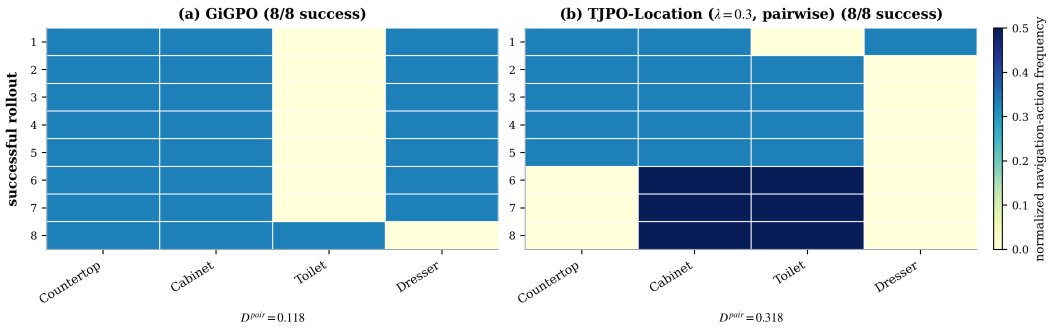


Figure 3: Location-descriptor view of the main-paper route case study for the prompt "put a spray-bottle in cabinet." Each row is one successful rollout, each column is a location activated by at least one rollout, and color denotes the normalized frequency of navigation actions targeting that location. Rows are ordered by their projection onto the first principal component of the pooled GiGPO and TJPO descriptor vectors, and both panels use a shared color scale.

where $K_{ij} = \exp(-\|z_i - z_j\|_2^2/10)$ and $\eta = 10^{-4}$. Here, n is the number of successful trajectories, not the descriptor dimension. Groups with fewer than two successes are assigned zero. The transformation is applied before aggregation and is used only for evaluation; it does not change the training-credit definition.

F Location-Diversity Case Study

Alternative Navigation Routes. Figure 3 provides the descriptor-level view of the same eight-rollout comparison visualized as route graphs in the main paper. The GiGPO rows are nearly identical: seven rollouts allocate equal navigation frequency to Countertop, Dresser, and Cabinet, reflecting the dominant Countertop–Dresser–Cabinet route. In contrast, the TJPO descriptors separate into multiple patterns, including rollouts that navigate directly through the Toilet and rollouts that retain the Dresser alternative. This greater between-rollout heterogeneity raises $D_{\text{loc}}^{\text{pair}}$ from 0.118 to 0.318. Together, the route graph and heatmap show that the increase comes from distinct successful navigation strategies rather than a change in task success; both methods solve all eight rollouts.

Figure 4 exposes where the trajectory diversity arises. For GiGPO, seven of eight rollouts execute the same six-action sequence through the Countertop and Dresser; the remaining rollout replaces the Dresser with the Toilet but otherwise follows the same plan, yielding two unique route signatures. TJPO instead produces three successful strategy groups: three five-action rollouts navigate directly to the Toilet, four six-action rollouts inspect the Countertop before using the Toilet, and one rollout retains the Dresser route.

The variation is localized to the search and navigation prefix. After a spray bottle is found, all rollouts use the same task-completion structure: take the bottle, navigate to the Cabinet, open it, and place the bottle inside. Thus, TJPO changes which successful route is selected without disrupting the reliable manipulation sequence. The three direct-Toilet trajectories are also one action shorter than the GiGPO trajectories, and none of the TJPO rollouts revisits a location. All 16 displayed trajectories contain zero invalid actions and zero location revisits. The increase in $D_{\text{loc}}^{\text{pair}}$ therefore reflects alternative efficient routes rather than longer or repetitive exploration.

Alternative Retrieval Locations before Cleaning. Figure 5 provides an additional location-diversity example for "clean some ladle and put it in diningtable." GiGPO follows the same six-action sequence in all eight rollouts: retrieve a ladle from the DiningTable, clean it at the SinkBasin, and return it to the DiningTable. TJPO follows this branch in six rollouts, while two rollouts retrieve a different ladle from a Countertop before completing the same cleaning and placement operations. Both methods succeed in all eight rollouts. The location-diversity score increases from 0 to 0.281, while the mean action count changes from 6.00 to 6.25. This example illustrates variation in retrieval locations while preserving the required processing and final placement operations.

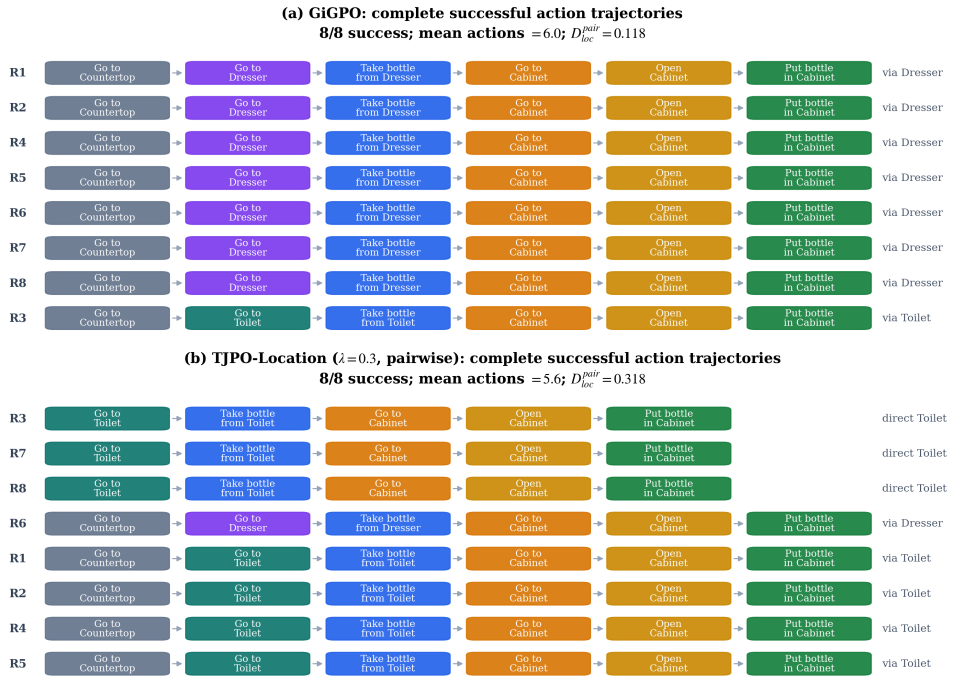


Figure 4: Complete successful environment-action trajectories for the same "put a spraybottle in cabinet" case. All eight sampled rollouts are shown for each method; no trajectories are omitted or selected. Rollouts are grouped by navigation strategy while retaining their original rollout identifiers. Object instance indices are omitted for readability, but every environment action is shown. Colors distinguish navigation destinations and task-completion operations.

G Practical Benefits of Behavioral Diversity

Evaluation suite. Table 5 reports results on a structurally verified subset of ALFWorld test games, containing 15 tasks from 14 scenes: nine Pick & Place, four Clean & Place, and two Cool & Place tasks. Each task admits successful solutions through two independent source location categories. We retain both intervention directions, yielding 30 cases for each of the User Constraints and Disruption conditions.

Task construction. Tasks are selected using PDDL, task metadata, and native environment replay, without using model scores. The two solution branches must use different target object instances. We exclude tasks with shared target objects, or with a forbidden location sharing its symbolic position with an unrestricted named receptacle. Source categories must not overlap the final goal category or required processing tools. Each branch must avoid the other source category, and the lengths of their verified plans must differ by at most two actions. These are feasible witness lengths, not claimed shortest-path lengths.

For each source category X, the intervention covers all initially navigable named locations assigned to that category. We verify successful execution using X, avoiding X in the original environment, avoiding X under both intervention conditions, and completing an alternative plan after one rejected access to X. Every successful replay must confirm target pickup and final placement within 50 steps. Both directions must pass these checks; all admitted tasks and directions are retained in the evaluation.

Evaluation conditions. *Standard* uses the original task without additional instructions or changes to the environment.

The *User Constraints* condition adds an instruction to avoid locations of type X, explicitly lists the corresponding locations, and asks the agent to complete the original task. The instruction is repeated

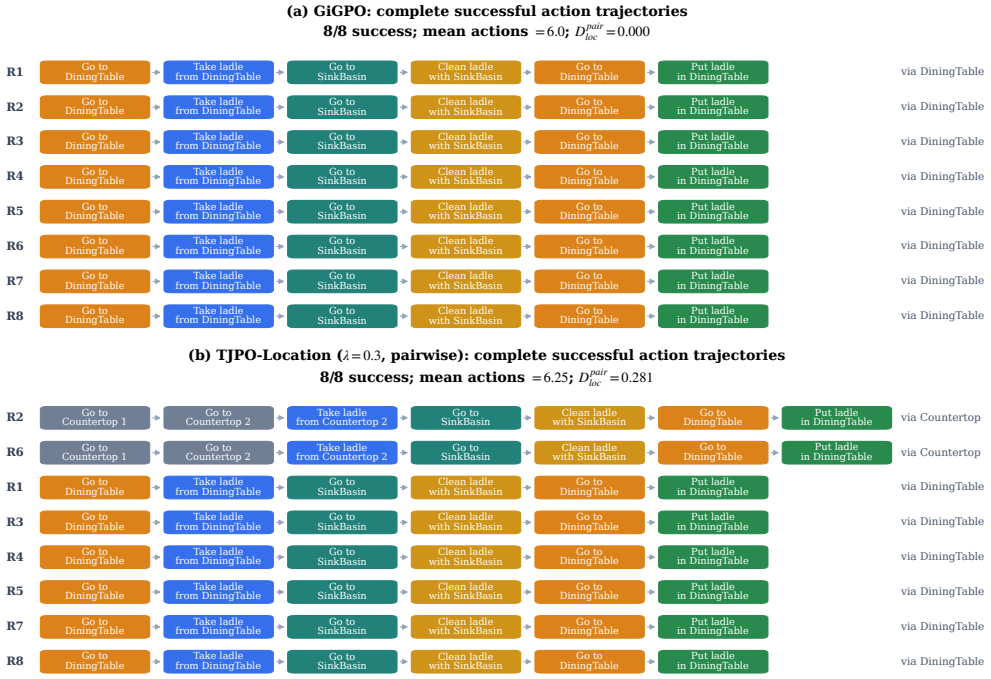


Figure 5: Complete successful trajectories for "clean some ladle and put it in diningtable." The top panel shows GiGPO and the bottom panel shows location-pairwise TJPO with $\lambda = 0.3$. All eight rollouts are retained for each method, with their original rollout identifiers preserved. Object instance indices and singleton-location indices are omitted for readability, while Countertop 1 and Countertop 2 are distinguished. Rows are grouped by navigation sequence; both panels use the same action-position scale, and no actions are omitted.

in each observation. This condition does not modify admissible actions or environment dynamics, so the agent can violate the requirement.

The *Disruption* condition makes the same predefined location set unavailable to both models from the start of the episode, without initially informing the agent. An attempted navigation or interaction involving this set is rejected and replaced by *look*, preserving the manipulation state while consuming one interaction step. The first rejection reveals the disruption. Subsequent observations and admissible actions reflect the persistent unavailability of these locations. Neither intervention supplies an alternative location or action plan.

Models and sampling. We compare GiGPO with location-pairwise TJPO and $\lambda = 0.5$. No additional training is performed for these interventions. Each case receives eight stochastic rollouts at temperature 0.4 with a maximum of 50 interaction steps. The environment seed is fixed to zero. For each model, the evaluation therefore contains 120 Standard, 240 User Constraints, and 240 Disruption trajectories.

Metrics and aggregation. Task success is determined by the native environment. Standard Success is the proportion of successful episodes without intervention. User Constraints Success requires both task completion and no environment-confirmed visit to the forbidden location set. Disruption Success is the proportion of successful episodes under location unavailability, including episodes that never attempt to access a forbidden location.

H Broader Impact

This work studies how to explicitly define and optimize trajectory diversity in RL-based post-training for LLM agents. A potential positive impact is improved controllability and robustness: by encouraging multiple high-quality behavioral modes, agents may become less dependent on a single brittle solution path and better adapt to changes in tools, interfaces, or environments. The use of user-specified, task-specific descriptors also makes the diversity objective more interpretable, allowing practitioners to specify which forms of behavioral variation are desirable for a given application.

At the same time, explicit diversity should be used together with appropriate safety constraints and task validation. In applications involving external tools or real-world operations, practitioners should ensure that diversified behaviors remain safe, reliable, and aligned with intended use.