WHICH STATES SHOULD BE COMPARED? BEHAVIORAL QUOTIENTS FOR INTERACTIVE POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative policy optimization converts rewards into advantages by normalizing within a reference group. For interactive agents, the appropriate reference group is unclear: exact observation identity splits situations with the same observed behavior, while broad pooling can merge distinctions that matter for control. We study this choice in graph-based path credit. From the action-labeled transition graph of a rollout batch, *Behavioral Quotient Policy Optimization* (BQPO) uses stable partition refinement to build a terminal-preserving quotient without parsing text, pixels, or task fields. With the unit edge costs used in our experiments, the quotient preserves observed success reachability, shortest-path credit, and action ordering. It therefore changes the group used for normalization, not the raw path target. Because equal path values do not make all samples interchangeable, an exact-first router keeps exact groups when local evidence identifies a clear continuation. It uses quotient groups when exact evidence is insufficient and, for other ambiguous cases, when a nontrivial class supplies complete action comparisons. On WebShop, ALFWorld, and Sokoban, BQPO improves the corresponding GraphGPO result in all seven reported aggregate comparisons, covering 1.5B and 7B language models and a 3B vision-language model.

1 INTRODUCTION

Foundation models increasingly act through tools, websites, games, and embodied environments rather than producing a single response (Schick et al., 2023; Yao et al., 2022b;a; Shridhar et al., 2021; Wang et al., 2023). In these settings, each action changes the evidence available to later decisions, while supervision often arrives only as a terminal outcome. A single success or failure signal must therefore be distributed over a sequence of heterogeneous decisions, each made under a different observation and a different set of admissible actions. Assigning that outcome to the decisions that produced it is thus a central obstacle to reinforcement learning for interactive agents.

Recent critic-free methods learn from relative outcomes without fitting a separate value model. GRPO compares responses to the same prompt (Shao et al., 2024; Yu et al., 2025), while multi-turn extensions introduce step-level or hierarchical groups (Wang et al., 2025; Feng et al., 2025; He et al., 2026). GraphGPO reconnects rollout transitions and propagates success through shortest paths to obtain step-level graph rewards (Cheng et al., 2026). A raw step reward, however, becomes an advantage only after it is centered and scaled within a reference group. The same reward can therefore produce a different update under a different comparison set. Existing methods increasingly refine *where* credit is assigned, but largely leave a second choice implicit: *against which experiences should that credit be compared?*

Responses to one prompt form a natural single-turn reference group. Interactive trajectories have no analogous unit: different-looking observations may encode the same control situation, while similar ones may require different actions. A product page revisited with a different scroll history, for example, may admit exactly the same meaningful choice, whereas two nearly identical pages may differ in one decisive detail. Exact identity yields small or singleton groups; broad pooling compares incompatible decisions. Parsers, task fields, and embedding thresholds introduce their own semantics

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.

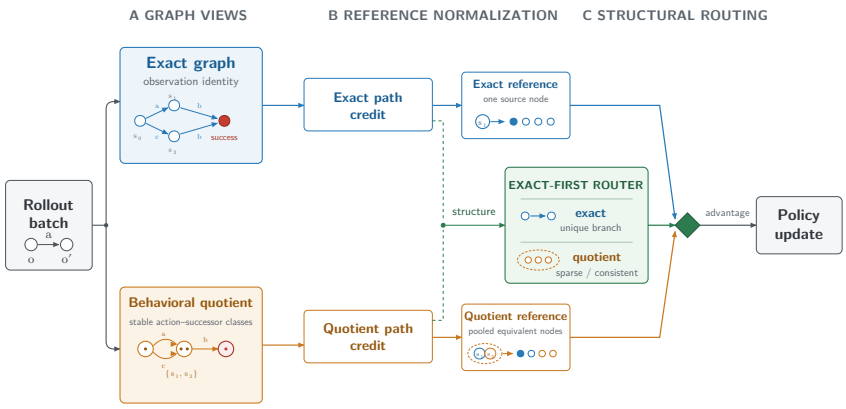

Figure 1: BQPO forms a stable quotient that preserves unit-cost path credit, then routes each transition between exact and quotient reference normalization.

or geometry. In an interactive graph, the state definition determines which transitions contribute to one another's advantage.

We define the comparison unit from behavior observed in the rollout graph. Two visited observations stay in the same class only if they have the same terminal type and, recursively, the same action-labeled successor classes. This follows the preservation view used in state abstraction and bisimulation (Givan et al., 2003; Li et al., 2006). Stable partition refinement finds the coarsest such partition of the observed graph. Every exact path then maps to a quotient path, and every quotient path can be lifted from any member of its source class. With unit edge costs, this preserves observed success reachability, distance, graph reward, and action ordering. The quotient can thus remove observation distinctions without changing path credit.

The quotient preserves path credit, but the best normalization group still depends on local evidence. Pooling adds comparison samples, while exact groups retain visit-specific support. BQPO keeps the exact reference when an observed continuation is uniquely supported. It uses the quotient when the exact readout cannot distinguish the transition, and in other ambiguous cases when a nontrivial quotient class has complete, matching action contrasts.

More broadly, the rollout graph can determine which state distinctions path credit needs to retain, while local evidence determines which states should share a normalization group. Our contributions span the method, its guarantees, and its evaluation:

- **Method.** We identify the comparison group as a separate modeling choice in interactive group-relative learning and propose BQPO, which builds a terminal-preserving behavioral quotient of the rollout graph by stable partition refinement and routes each transition between exact and quotient reference groups, without a task parser, similarity threshold, or mixture coefficient.
- **Theory.** We prove that refinement yields the coarsest stable partition of the observed graph (Lemma 4.1) and that, under unit edge costs, the quotient preserves success reachability, shortest-path distance, graph rewards, and all observed action orderings (Theorem 4.2). A bias–variance analysis further characterizes when pooling improves estimation of the normalization mean (Proposition 4.3).
- **Experiments.** On WebShop, ALFWorld, and Sokoban, BQPO improves the corresponding GraphGPO result in all seven aggregate comparisons, covering 1.5B and 7B language models and a 3B vision-language model. Structural diagnostics confirm nontrivial graph compression and sustained use of both reference granularities.

2 Setup: Graph Credit from Finite Rollouts

The comparison group can be studied only after specifying what is being compared. We start from the finite transition graph of a rollout batch and a path-credit readout, then make the state-to-group relation explicit.

2.1 THE EVIDENCE AVAILABLE IN ONE ROLLOUT BATCH

For task instance x, a policy π_θ interacts with an environment for at most T steps. At step t, it receives observation o_t, samples action $a_t \sim \pi_\theta(\cdot \mid o_t, x)$, and receives the next observation o_{t+1}. A rollout batch supplies a finite empirical action-labeled observation transition system

$$\widehat{\mathcal{T}} = (\widehat{\mathcal{O}}, \mathcal{A}, \widehat{\mathcal{E}}, \widehat{\tau}), \qquad \widehat{\mathcal{E}} \subseteq \widehat{\mathcal{O}} \times \mathcal{A} \times \widehat{\mathcal{O}}, \tag{1}$$

where $\widehat{\tau}(o) \in \{\text{nonterminal}, \text{success}, \text{failure}\}$ is the observed terminal type. This system contains only transitions from the current rollout batch; it is not a learned model or the underlying MDP. We call o an observation rather than a state because a web page, textual scene, or rendered board may omit relevant history. All structural claims below concern $\widehat{\mathcal{T}}$, not unobserved environment dynamics.

2.2 PATH CREDIT ON A ROLLOUT GRAPH

We use the shortest-path readout of GraphGPO (Cheng et al., 2026) so that the effect of the comparison group can be identified directly. Given a node map ϕ, rollout transitions induce a directed graph whose nodes are the attained values of ϕ. Let s denote one such node, $\mathcal{S}^+$ the set of observed successful terminal nodes, and $c(e) > 0$ the cost of edge e. If $\mathcal{P}(s, \mathcal{S}^+)$ is the set of directed paths from s to any node in $\mathcal{S}^+$, including the zero-length path when $s \in \mathcal{S}^+$, the empirical distance is

$$d(s) = \begin{cases} \min\limits_{p \in \mathcal{P}(s,\mathcal{S}^+)} \sum\limits_{e \in p} c(e), & \mathcal{P}(s, \mathcal{S}^+) \neq \varnothing, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2}$$

Thus distance is zero at a successful terminal and finite exactly when observed experience contains a route to success. For an action a observed at s, the associated one-step cost-to-success and transition reward are

$$Q(s, a) = \min_{(s,a,s') \in \mathcal{E}} \{c(s, a, s') + d(s')\}, \tag{3}$$

$$R^G(s, a, s') = r_+ \, \omega^{d(s')}, \tag{4}$$

with $r_+ > 0$ and $0 < \omega < 1$. Thus minimizing Q and maximizing R^G induce the same ordering over observed routes under the unit edge costs used in our experiments. For numerical evaluation, GraphGPO replaces $+\infty$ by $d_{\max} + 1$, where $d_{\max}$ is the largest finite distance in the graph and is set to zero if that set is empty. The quotient preserves both $d_{\max}$ and this surrogate, as shown in Theorem 4.2. The central question is then which transitions should share the mean and scale that turn these path rewards into relative advantages.

2.3 A STATE IS A CREDIT-SHARING DECISION

Let $\phi(o)$ map an observation to a graph node. This map serves two roles. It defines the transition graph on which d and R^G are computed, and it groups all occurrences with the same $\phi(o)$ when their rewards are normalized. A quotient can preserve the first role while changing the second.

We therefore seek a stable equivalence relation that removes observation distinctions without changing empirical path credit. Its classes are the largest groups allowed by observed action-support stability. They are not claims about latent MDP states: the relation ignores transition counts, hidden variables, and unseen actions. A router then decides when this behavioral class—which may still be a singleton—should replace exact identity as the normalization reference.

3 BEHAVIORAL QUOTIENTS AS CREDIT REFERENCE CLASSES

We first construct a quotient from observed action-successor support, then define the graph readouts and the rule selecting their normalization groups. Section 4 shows why the quotient preserves path credit and when larger groups improve estimation. Together these steps define Behavioral Quotient Policy Optimization (BQPO).

3.1 TWO GRAPHS FROM ONE ROLLOUT BATCH

The exact graph $G_{\mathrm{E}} = (V_{\mathrm{E}}, E_{\mathrm{E}})$ groups observations only when they are equal. A stable hash may index them in code, but does not introduce a similarity rule.

The abstract graph G_{A} is a quotient of this exact graph. We first separate successful, failed, and nonterminal nodes by setting

$$\ell_0(s) = \mathrm{terminal_type}(s). \tag{5}$$

At iteration $k+1$, the signature of a node contains its current class and the observed pairs of action labels and successor classes:

$$\ell_{k+1}(s) = \mathrm{ID}(\ell_k(s), \{(a, \ell_k(s')) : (s, a, s') \in E_{\mathrm{E}}\}), \tag{6}$$

Let K be the first iteration at which the induced partition no longer changes. The quotient map and quotient edges are then

$$\phi_{\mathrm{A}}(s) = \ell_K(s), \qquad V_{\mathrm{A}} = \{\phi_{\mathrm{A}}(s) : s \in V_{\mathrm{E}}\}, \qquad E_{\mathrm{A}} = \{(\phi_{\mathrm{A}}(s), a, \phi_{\mathrm{A}}(s')) : (s, a, s') \in E_{\mathrm{E}}\}. \tag{7}$$

The map ID gives equal signatures the same deterministic identifier. Since each new signature includes the previous label, refinement can split classes but never merge them, and must stop on a finite graph. Nodes with the same final label form one quotient node. They remain together only if terminal type and every finite unfolding of observed action-labeled successors fail to distinguish them.

An equivalent view unfolds the observed action-labeled branches from each node: the depth-k label records one more successor level than depth $k-1$, and any difference propagates backward until the nodes split. This condition is stronger than equality of a single shortest-path value—which is what makes path lifting possible—and uses no visual or textual similarity.

We call this an *empirical action-labeled support quotient*. Relabeling exact nodes does not change it as long as terminal types and action-labeled edges stay fixed. The construction is related to partition refinement and bisimulation (Givan et al., 2003; Li et al., 2006), but is weaker than stochastic MDP bisimulation: it discards transition counts and probabilities, uses rewards only through terminal type, and makes no claim about unseen dynamics.

3.2 READOUTS ON BOTH PARTITIONS

We apply (2)–(4) to both graphs, obtaining $Q_{\mathrm{E}}, R_{\mathrm{E}}$ and $Q_{\mathrm{A}}, R_{\mathrm{A}}$. Transitions are aggregated by node identity, and shortest paths to observed success provide edge-level credit. With unit edge costs, Theorem 4.2 shows that the raw rewards agree on every observed transition:

$$R_{\mathrm{E}}(s, a, s') = R_{\mathrm{A}}(\phi_{\mathrm{A}}(s), a, \phi_{\mathrm{A}}(s')). \tag{8}$$

Their normalized advantages can still differ because their source groups can differ. The separate notation keeps this distinction explicit and matches the implementation.

For a quotient class z, let

$$M_z = \{s \in V_{\mathrm{E}} : \phi_{\mathrm{A}}(s) = z\} \tag{9}$$

be its exact members. Throughout, actions are keyed by their environment-provided labels. Thus two action-order signatures are equal only if their alternative-action keys also match. This avoids a hidden task parser, at the cost of treating differently named but semantically equivalent actions as different keys.

3.3 SELECTING THE REFERENCE GRANULARITY

For transition $i = (s_i, a_i, s_i')$, define $\mathrm{disc}_{\mathrm{E}}(i) = 1$ precisely when the exact source and successor have finite readouts, the observed edge and action score exist, and the exact best action is unique. The first route is then

$$m_i^{\mathrm{cov}} = 1 - \mathrm{disc}_{\mathrm{E}}(i). \tag{10}$$

When $m_i^{\mathrm{cov}} = 1$, routing selects the quotient reference group, which contains the exact group. Pooling can change a normalized advantage when class members assign different frequencies to the same reward values. It does not add a path missing from the exact support graph; path lifting rules

this out. The implementation calls this the "coverage" route because it expands the comparison set, not graph reachability.

Suppose instead that $\mathrm{disc}_{\mathrm{E}}(i) = 1$. The router first tests whether the observed successor is the unique lexicographic maximizer for the selected action under (edge support, incoming support, negative distance), where the two supports count occurrences of that action-labeled edge and of all edges entering the candidate successor, respectively. Let $\mathrm{local}_{\mathrm{E}}(i)$ denote this predicate. If it is true, the observed transition has a uniquely supported exact successor and exact credit is retained. This deterministic, threshold-free predicate protects well-supported exact decisions. For notational completeness, set $\mathrm{local}_{\mathrm{E}}(i) = 0$ whenever $\mathrm{disc}_{\mathrm{E}}(i) = 0$.

When $\mathrm{local}_{\mathrm{E}}(i) = 0$, the second route requires complete action comparisons across a nontrivial quotient class. For any score map q whose domain contains a and at least one alternative, define the keyed contrast signature

$$C_q(a) = \big\{\big(b, \mathrm{sign}[q(a) - q(b)]\big) : b \in \mathrm{dom}(q) \setminus \{a\}\big\}. \tag{11}$$

Because Q is a cost, signs $-1, 0, +1$ mean that a is respectively better than, tied with, or worse than b. The signature discards numerical margins but retains alternative-action identities. Write $C_s(a) = C_{Q_{\mathrm{E}}(s,\cdot)}(a)$ and $C_z(a) = C_{Q_{\mathrm{A}}(z,\cdot)}(a)$. For $z = \phi_{\mathrm{A}}(s_i)$, let $\mathrm{cons}_z(a_i)$ be true exactly when (i) $|M_z| \geq 2$, (ii) every member signature $C_s(a_i)$ is defined, (iii) these signatures are identical over all $s \in M_z$, and (iv) the quotient signature is identical to them. The consistency route is

$$m_i^{\mathrm{con}} = \mathrm{disc}_{\mathrm{E}}(i)\big(1 - \mathrm{local}_{\mathrm{E}}(i)\big)\mathbb{I}\{\mathrm{cons}_z(a_i)\}. \tag{12}$$

Under exact fixed-point refinement and unit costs, equality of defined contrasts follows from Theorem 4.2. The explicit check makes this comparison well-defined in finite batches: the route requires a nontrivial class and complete contrasts for all members. The two routes are mutually exclusive. Their union gives the final mask

$$m_i = m_i^{\mathrm{cov}} \vee m_i^{\mathrm{con}}. \tag{13}$$

The rule keeps the exact reference when one successor has unique local support, uses the quotient when the exact readout cannot discriminate, and uses it for other ambiguous cases when a nontrivial class supplies complete action contrasts.

3.4 Separate normalization and hard selection

The mask in Equation (13) is fixed before normalization. The exact and quotient graphs can have different source groups even though their raw unit-cost rewards agree by Equation (8). Let $g_r(i)$ be the source group of transition i under representation $r \in \{\mathrm{E}, \mathrm{A}\}$, with group mean μ_r and standard deviation σ_r. We standardize each branch within its own groups:

$$Z_{\mathrm{E}}(i) = \frac{R_{\mathrm{E}}(i) - \mu_{\mathrm{E}}(g_{\mathrm{E}}(i))}{\sigma_{\mathrm{E}}(g_{\mathrm{E}}(i)) + \epsilon}, \tag{14}$$

$$Z_{\mathrm{A}}(i) = \frac{R_{\mathrm{A}}(i) - \mu_{\mathrm{A}}(g_{\mathrm{A}}(i))}{\sigma_{\mathrm{A}}(g_{\mathrm{A}}(i)) + \epsilon}. \tag{15}$$

This is the usual group-relative z-score. Here it turns each structural class into a comparison group. Since the mask is chosen first, reward scale cannot change the routing decision. The selected graph advantage is

$$A_i^{\mathrm{BQPO}} = (1 - m_i)Z_{\mathrm{E}}(i) + m_i Z_{\mathrm{A}}(i). \tag{16}$$

This selected advantage enters the policy update alongside the episode advantage, KL term, clipped loss, and balancing weights used in GraphGPO. Unlike soft mixing, Equation (16) introduces no graph-mixture coefficient η. The complete execution order appears in Algorithm 1.

4 What the Quotient Preserves—and What It Changes

The quotient need not recover the environment's hidden state. What matters here is whether it changes the path credit used for training. We answer this question on the finite rollout graph.

Lemma 4.1 (Stable empirical support partition)**.** *At convergence of Equation* (6)*, any two nodes in the same quotient class have the same terminal label and the same set of action-labeled successor classes. Moreover, the resulting partition is the coarsest refinement of the initial terminal partition with this property.*

The proof is deferred to Appendix A.4.

Theorem 4.2 (Path and readout preservation)**.** *Let $\pi(s) = [s]$ be the quotient map induced by the stable partition. For every state s, action word $w = (a_1, \dots, a_k)$, and terminal type y, there is a w-labeled path from s to a node of type y if and only if there is such a path from $\pi(s)$ in the quotient. Hence, with unit edge costs,*

$$d_{\mathrm{E}}(s) = d_{\mathrm{A}}(\pi(s)), \qquad Q_{\mathrm{E}}(s, a) = Q_{\mathrm{A}}(\pi(s), a) \tag{17}$$

for every observed action, including the value $+\infty$. Consequently the raw transition rewards and all observed action orderings are preserved.

The proof in Appendix A.4 is constructive: exact paths project to the quotient, and stability lifts each quotient edge from any representative. The partition is therefore stronger than one designed only to preserve a particular shortest-path value. Its guarantee concerns the observed graph.

Since the raw rewards agree, any change in training signal must come from the normalization group. If I_s indexes transition occurrences with exact source s, then the quotient reference set is the disjoint union

$$I_z = \bigcup_{s \in M_z} I_s, \qquad |I_z| = \sum_{s \in M_z} |I_s|. \tag{18}$$

Thus quotienting leaves each raw reward fixed but can change the empirical mean and scale against which that reward is compared.

Stability preserves the set of raw reward values available at every member of a quotient class. A constant-reward exact group therefore stays constant after pooling. The effect of a larger group comes from different frequencies of existing reward values, not from creating a new reward contrast.

An example shows how this change reaches a selected transition. Consider two sources in one stable class, both with edges $a \to u$, $a \to v$, and $b \to v$, where $d(u) < d(v) < \infty$ gives raw rewards $H > L$. Across four occurrences, the first source has two $a \to u$, one $a \to v$, and one $b \to v$; the second has one $a \to u$, two $a \to v$, and one $b \to v$. Both have identical support, and a is uniquely best by Q. At the second source, the $a \to u$ occurrence has less edge support than $a \to v$, so the local test defers to the quotient. With complete two-action contrasts, the consistency route selects it. Ignoring ϵ and using population standard deviations, its exact z-score is $\sqrt{3}$ and its quotient z-score is $\sqrt{5/3}$. The path reward remains H.

Proposition 4.3 (Benefit and cost of pooling)**.** *Suppose reference observations are mutually independent and, for every exact state s in a quotient class, have the form $X = \mu_s + \varepsilon$, with $\mathbb{E}[\varepsilon] = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$. Let $n_s = |I_s|$, let $n_z = |I_z|$, and let $\bar{\mu}_z = \sum_{s \in M_z} n_s \mu_s / n_z$. The exact sample mean has mean-squared error σ^2/n_s for μ_s, whereas the pooled mean has*

$$\mathrm{MSE}_s(\widehat{\mu}_z) = \frac{\sigma^2}{n_z} + (\bar{\mu}_z - \mu_s)^2. \tag{19}$$

Pooling is therefore preferable for this baseline exactly when its squared class heterogeneity is smaller than the variance reduction $\sigma^2(1/n_s - 1/n_z)$.

This calculation captures the tradeoff behind the method: quotienting can raise n_z without changing path value, reducing the noise of a group mean, while heterogeneous exact-state means can offset that gain. This motivates retaining well-supported exact distinctions. The proposition analyzes the normalization mean; the router operationalizes the choice using finite-graph evidence. In the experiments, compression measures available pooling and the routing trace records which reference is selected.

5 Experimental Evaluation

The experiments test the two predictions of our design: BQPO should improve learning across tasks and model sizes, and its quotient should create larger reference groups in practice. We report task

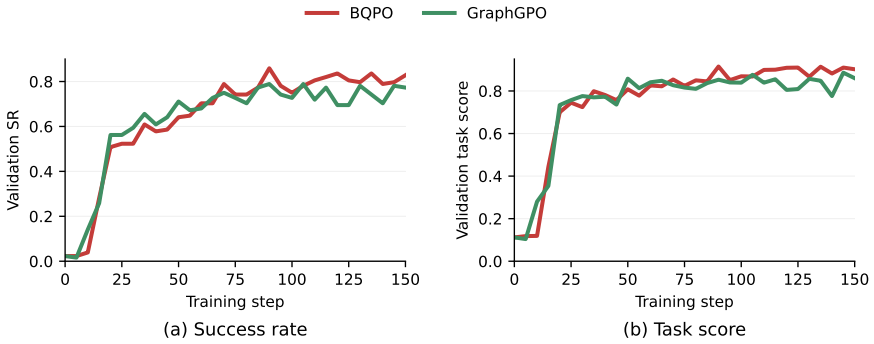

Figure 2: WebShop validation dynamics for one matched Qwen2.5-1.5B-Instruct run.

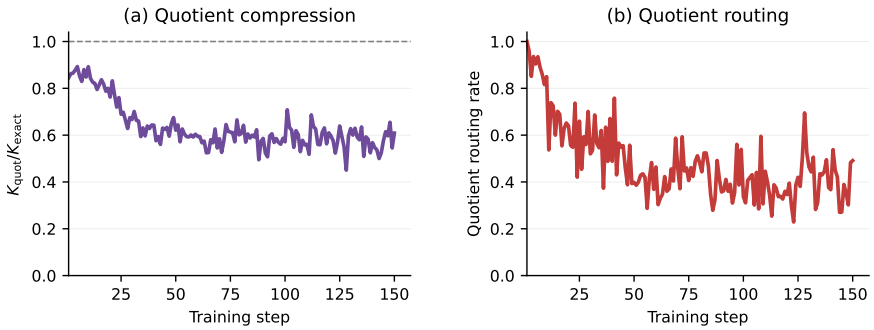

Figure 3: Structure of the matched WebShop run in Figure 2. **Left:** quotient-to-exact visited-node ratio; values below one indicate pooling. **Right:** fraction of transitions assigned the quotient reference.

performance first, then examine the training and structural traces that explain how the two reference granularities are used.

5.1 Experimental design

We use WebShop (Yao et al., 2022a), ALFWorld (Shridhar et al., 2021), and Sokoban (Schrader, 2018), which respectively exercise shopping from web text, instruction following in a textual household environment, and visual box-pushing. Text runs use Qwen2.5-1.5B-Instruct or Qwen2.5-7B-Instruct (Qwen, 2025); visual runs use Qwen2.5-VL-3B-Instruct (Bai et al., 2025).

For a controlled comparison, we keep GraphGPO's prompts and environment interfaces (including its augmented ALFWorld observations and the text-rich WebShop interface), 150 optimization steps, eight rollouts per group, 128 parallel environments, and unit graph-edge costs. Episodes permit up to 50 interactions in ALFWorld and 15 in WebShop and Sokoban. The corresponding graph-distance discounts are 0.10, 0.20, and 0.80. Optimizer, decoding, reward, and system settings are also held fixed; Table 3 records the key settings, and the full base protocol is documented by GraphGPO (Cheng et al., 2026). The method-specific choices are exact observation identity without goal conditioning, the quotient in Equation (6), the gate in Equation (13), and separate branch normalization with hard selection and no soft mixture term.

ALFWorld and Sokoban are evaluated by success rate (SR); WebShop uses SR and its graded task score. Each reported BQPO entry is the mean and sample standard deviation across three independently trained runs. ALFWorld "All" denotes the reported aggregate SR over its six task categories.

Table 1: Test performance on ALFWorld and WebShop. ALFWorld columns report success rate (%) by task type and the reported aggregate (All); WebShop reports task score and success rate (%). RL-trained results are averaged over three random seeds, and the best result within each model block is bold.

Type	Method	ALFWorld							WebShop	
		Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Succ.
Closed-Source Models										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training	PPO	64.8(3.5)	40.5(6.9)	57.1(4.9)	60.6(6.6)	46.4(4.0)	47.4(1.9)	54.4(3.1)	73.8(3.0)	51.5(2.9)
RL Training	RLOO	88.3(3.0)	52.8(8.6)	71.0(5.9)	62.8(8.7)	66.4(5.5)	56.9(4.7)	69.7(2.5)	73.9(5.6)	52.1(6.7)
RL Training	GRPO	82.89(3.62)	82.14(6.37)	73.86(6.84)	78.57(0.00)	77.78(4.54)	71.43(3.89)	77.86(1.33)	84.73(0.49)	71.35(2.05)
RL Training	GiGPO	98.81(1.68)	95.16(3.89)	81.46(0.56)	78.57(0.00)	94.44(0.00)	93.65(5.94)	90.88(0.97)	87.94(0.43)	73.83(2.30)
RL Training	GraphGPO	95.15(1.62)	**100.0(0.00)**	85.26(2.58)	85.71(5.83)	**96.30(2.61)**	93.65(2.24)	92.71(1.32)	89.29(1.48)	78.65(3.86)
RL Training	**BQPO**	**98.97(1.79)**	**100.0(0.00)**	**91.20(3.98)**	**89.77(9.31)**	93.80(0.00)	**95.20(4.57)**	**94.82(0.83)**	**90.30(2.45)**	**81.00(4.57)**
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	92.3(4.0)	64.0(8.4)	92.5(2.4)	89.5(7.0)	80.3(2.0)	68.8(8.3)	80.4(2.7)	81.4(3.1)	68.7(5.1)
RL Training	RLOO	87.6(4.3)	78.2(8.3)	87.3(5.8)	81.3(7.6)	71.9(5.2)	48.9(8.4)	75.5(4.6)	80.3(3.2)	65.7(4.0)
RL Training	GRPO	88.98(5.30)	91.98(4.43)	77.89(4.58)	78.57(0.00)	90.74(5.24)	71.43(3.89)	83.33(2.05)	84.31(1.27)	75.00(2.78)
RL Training	GiGPO	97.53(1.75)	**100.0(0.00)**	83.98(1.32)	90.48(6.73)	**94.44(0.00)**	**100.0(0.00)**	94.27(1.33)	86.72(1.44)	78.38(1.94)
RL Training	GraphGPO	**100.0(0.00)**	**100.0(0.00)**	91.40(1.50)	92.86(5.83)	**94.44(0.00)**	92.06(2.24)	95.31(1.10)	86.94(0.68)	80.31(1.33)
RL Training	**BQPO**	**100.0(0.00)**	**100.0(0.00)**	**95.37(4.23)**	**100.0(0.00)**	89.80(6.93)	98.57(2.48)	**97.29(0.93)**	**89.97(0.81)**	**81.27(2.06)**

Table 2: Sokoban 6×6 test SR (%) with Qwen2.5-VL-3B-Instruct.

Type	Method	SR
Prompting	Qwen2.5-VL	11.7
RL Training	GRPO	67.1
RL Training	GiGPO	76.92
RL Training	GraphGPO	86.98(0.73)
RL Training	**BQPO**	**87.77(1.22)**

5.2 Main results

Table 1 places our BQPO runs alongside the baselines reported by GraphGPO (Cheng et al., 2026) for the corresponding models and benchmarks. It reports means and sample standard deviations across independent runs, omitting uncertainty where the source did not report it.

Overall results. BQPO improves all seven aggregate metrics relative to GraphGPO: ALFWorld SR, WebShop task score, and WebShop SR at both text-model sizes, plus Sokoban SR. The gains are 2.11/1.98 points on ALFWorld, 1.01/3.03 on WebShop score, and 2.35/0.96 on WebShop SR. Sokoban rises from 86.98 to 87.77, extending the result to visual spatial planning. These gains build on a strong graph-credit baseline: GraphGPO already outperforms GiGPO and GRPO in most settings and reaches over 92% ALFWorld SR at both text-model sizes.

WebShop's two metrics are complementary: SR requires completing the requested task, whereas task score also rewards partial progress. BQPO improves both at both model sizes, so the gain is not obtained by trading completion rate for attempt quality. Together with the ALFWorld and Sokoban results, this agreement shows that changing the reference group is effective across different observation spaces and reward summaries.

The ALFWorld gains recur in the same categories at both model sizes: Cool ($+5.94$ and $+3.97$), Look ($+4.06$ and $+7.14$), and Pick2 ($+1.55$ and $+6.51$). These tasks involve repeated decisions across rooms and receptacles, where the same observed action structure can occur under different textual observations. Heat is the repeated exception. This variation motivates the router's per-transition choice between exact evidence and quotient pooling.

5.3 Training dynamics and structural behavior

The matched WebShop run adds the training-time view missing from endpoint scores: in Figure 2, the advantage persists over steps 90–150, where mean validation SR and task score increase by 6.0 and 5.0 points.

Figure 3 shows that the quotient creates larger groups while the router continues to use both branches. The quotient has 63.3% as many visited source nodes as the exact graph on average, and 58.1% over steps 90–150. Quotient routing averages 48.4% overall and 39.6% late in training. It begins near one and declines over training. The graph can therefore become more compact even as the router uses exact groups more often. By Theorem 4.2, this compression changes reference groups rather than path lengths.

5.4 Ablation evidence

The WebShop ablations in Appendix B separate the value of pooling from the structure that makes it useful. Always using the behavioral quotient reaches 79.93% SR and 89.97 task score; full BQPO reaches 81.00% and 90.30. A random partition matched to the quotient's class sizes instead leaves validation success near zero. Together, these controls show that larger groups are useful when they preserve observed decision structure, and that per-transition routing gives the highest reported means.

6 Related Work

Group-relative optimization. GRPO replaces a learned critic with comparisons among responses to the same prompt, and DAPO develops this recipe into a scalable training system (Shao et al., 2024; Yu et al., 2025). In interactive tasks the comparison unit is less obvious: rewards arrive after several environment transitions, and distinct rollouts may revisit related states. Recent work addresses this setting through multi-turn rollout optimization, step-level groups, and hierarchies of comparison groups (Wang et al., 2025; Feng et al., 2025; He et al., 2026). These choices specify how rewards are compared across turns; our question is which visited observations can serve as one reference class for a decision.

Credit assignment for agents. Interactive-agent training has used trajectory returns, decomposed actions, and frameworks that assign feedback across arbitrary agent executions (Wen et al., 2024; Luo et al., 2025). GraphGPO instead reconnects observed transitions into a graph and propagates terminal outcomes by shortest-path distance (Cheng et al., 2026). It supplies the path-credit readout used here. Our quotient shows that this readout can be preserved while the reference relation changes: repeated decision structure can be compared even when raw observations differ.

State abstraction. MDP homomorphisms, bisimulation, and related abstraction theories characterize when states can be merged while preserving rewards, transition laws, or optimal control (Givan et al., 2003; Li et al., 2006). Their usual object is a known or estimated MDP; our construction is instead built from the finite support of the current rollout graph, refining states by terminal labels and action-labeled successor blocks. The quotient is therefore neither an embedding cluster nor a recovered latent state: it supplies a behavior-based normalization group, while the policy still acts on the original observation and exact graph paths still define credit.

7 Conclusion

Interactive credit assignment depends on both how terminal outcomes reach individual decisions and which experiences those decisions are compared with. BQPO makes the second choice explicit: a stable behavioral quotient preserves observed unit-cost path credit, while local evidence selects when observations share a normalization group. It improves GraphGPO in all seven aggregate metrics across text and visual tasks and 1.5B, 3B, and 7B models. Together, the guarantee and results show that comparison groups can be redefined without changing the underlying path reward.

AI USE STATEMENT

Generative AI tools were used to improve the wording of the manuscript and to assist in checking the correctness of the proofs. The authors reviewed all AI-assisted suggestions, independently verified the final proofs, and take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

The graph construction, routing rule, and normalization procedure are specified in Section 3; assumptions and formal guarantees appear in Section 4, with complete proofs in Appendix A.4. Experimental protocols and principal implementation settings are reported in Section 5 and the appendix.

REFERENCES

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.

Xin Cheng, Shuo He, Lang Feng, HaiYang Xu, Ming Yan, Lei Feng, and Bo An. Beyond trajectory-level attribution: Graph-based credit assignment for agentic reinforcement learning. *arXiv preprint arXiv:2605.26684*, 2026.

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

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1–2):163–223, 2003.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2602.22817*, 2026.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

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

Qwen. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.

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

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.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

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.

Muning Wen, Ziyu Wan, Jun Wang, Weinan Zhang, and Ying Wen. Reinforcing llm agents via policy optimization with action decomposition. In *NeurIPS*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 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 *ICLR*, 2022b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

A Supplementary Technical Details

A.1 Full procedure

Algorithm 1 lists the full execution order. Graph structure determines the route before either branch is normalized; the policy update otherwise follows GraphGPO.

Algorithm 1 Behavioral Quotient Policy Optimization

Input: rollout transitions $\mathcal{D}$ and the GraphGPO readout
Construct exact graph G_{E} using observation equality
Refine labels by Equation (6); construct quotient G_{A}
Compute $(R_{\mathrm{E}}, Q_{\mathrm{E}})$ and $(R_{\mathrm{A}}, Q_{\mathrm{A}})$
for each observed transition $i = (s_i, a_i, s_i')$ **do**
  Evaluate $\mathrm{disc}_{\mathrm{E}}(i)$
  $m_i^{\mathrm{cov}} \leftarrow 1 - \mathrm{disc}_{\mathrm{E}}(i)$
  $m_i^{\mathrm{con}} \leftarrow 0$
  if $\mathrm{disc}_{\mathrm{E}}(i) = 1$ and $\mathrm{local}_{\mathrm{E}}(i) = 0$ **then**
    $z \leftarrow \phi_{\mathrm{A}}(s_i)$
    Compute $C_s(a_i)$ for all $s \in M_z$ and compute $C_z(a_i)$
    $m_i^{\mathrm{con}} \leftarrow \mathbb{I}\{\mathrm{cons}_z(a_i)\}$
  end if
  $m_i \leftarrow m_i^{\mathrm{cov}} \vee m_i^{\mathrm{con}}$
end for
Normalize R_{E} and R_{A} within their own source groups
Select A_i^{BQPO} by Equation (16); apply the GraphGPO update

Implementation notes. Both graphs use the same rollout batch, edge costs, and terminal outcomes. We run partition refinement to a fixed point, score each graph, and compute the route from the unnormalized structure. Exact and quotient rewards are then standardized within their own source-node groups and selected by the Boolean mask. Reward scale therefore cannot affect routing.

In all experiments, the fine graph uses exact observation identity without a goal-conditioned key. The coarse graph uses the support quotient in Equation (6), and both branches use z-score normalization. Only the coverage and consistency routes in Equation (13) select quotient credit; there is no continuous mixture term. All other choices follow the matched GraphGPO protocol in Section 5.

Table 3: Key task-specific settings inherited by all matched runs. Lengths are in tokens, H is the episode horizon, and ω is the graph-distance discount.

Benchmark	Model	Prompt	Response	H	ω
ALFWorld	Qwen2.5-1.5B/7B	2048	512	50	0.10
WebShop	Qwen2.5-1.5B/7B	5120	512	15	0.20
Sokoban	Qwen2.5-VL-3B	1024	512	15	0.80

All runs use 150 optimization steps, eight rollouts per group, 128 parallel environments, actor learning rate 10^{-6}, KL coefficient 0.01, and unit graph-edge costs. Rollout and validation temperatures are 1.0 and 0.4. Successful termination receives reward 10, ordinary failure receives 0, and an invalid action receives -0.1. The mini-batch size is 256 for ALFWorld and 64 for WebShop and Sokoban. Text agents retain the two most recent interaction steps in the prompt. All experiments are run on an NVIDIA A100 GPU cluster.

A.2 Boundary conventions

We use the following conventions for boundary cases. Successful terminal nodes admit the zero-length path and therefore have distance zero. An exact transition is not discriminative when its source or successor lacks a finite readout, its selected edge or action score is absent, or the best exact action is tied; these cases activate the coverage route. As established in Theorem 4.2, this route does not

create reachability. It changes the normalization reference used for the sample. Ties are otherwise retained explicitly through the zero entry of the sign function in Equation (11).

The consistency route is more restrictive. It is disabled for singleton quotient classes, for members lacking the selected action or an observed alternative, and whenever the keyed alternative-action sets differ. A class therefore cannot pass merely because the available numerical scores happen to be close. Finally, if every reward in a normalization group is identical, its centered values are zero and the numerical stabilizer in Equation (15) keeps the normalized branch finite. None of these conventions changes the mask after normalization.

A.3 Offline trace analysis

The archived WebShop trace behind Figures 2 and 3 also records reference-group sizes. At each optimization step, both summaries use the same N transition occurrences and report the average exact and quotient source-group sizes, N/K_{E} and N/K_{A}, together with the quotient routing rate. These sizes are averages across source groups. The node-ratio statistic in Figure 3 therefore has the equivalent group-mean form

$$\frac{K_{\mathrm{A}}}{K_{\mathrm{E}}} = \frac{N/K_{\mathrm{E}}}{N/K_{\mathrm{A}}}. \tag{20}$$

In particular, $N/K_{\mathrm{A}} > N/K_{\mathrm{E}}$ verifies that quotienting enlarges empirical reference groups on average across source groups.

Table 4: Reference-group sizes in the matched WebShop trace. Sizes count transition occurrences per source-reference group; expansion is the stepwise mean of $(N/K_{\mathrm{A}})/(N/K_{\mathrm{E}})$.

Window	Exact size	Quotient size	Expansion	Quotient route
Steps 1–150	3.75	6.09	1.61×	48.4%
Steps 90–150	3.89	6.73	1.73×	39.6%

The mean quotient group is larger at all 150 logged steps. Over the full run, its size is $1.61\times$ the exact-group size on average; over steps 90–150, the ratio rises to $1.73\times$ while quotient routing remains at 39.6%. Thus quotienting supplies larger empirical comparison sets throughout training, rather than only relabeling graph nodes.

A.4 Proofs of the structural properties

Proof of Lemma 4.1. Let $\mathcal{P}_k$ be the partition induced by ℓ_k. Because the old label $\ell_k(s)$ is included in the next signature, $\mathcal{P}_{k+1}$ can only split blocks of $\mathcal{P}_k$. Every strict refinement increases the number of blocks, so on a finite node set the sequence reaches a fixed point. Equality of the signatures at that point gives equality of terminal labels and of the sets of action-labeled successor classes.

It remains to establish coarseness. Let $\mathcal{P}'$ refine the terminal partition and be stable under the same action-labeled successor condition. We show inductively that $\mathcal{P}'$ refines $\mathcal{P}_k$. This is true for $k = 0$. If two nodes share a block of $\mathcal{P}'$, stability gives the same action-labeled successor blocks in $\mathcal{P}'$. Mapping those blocks to the coarser $\mathcal{P}_k$ blocks yields identical k-level signatures, so the two nodes also share a block of $\mathcal{P}_{k+1}$. Thus $\mathcal{P}'$ refines every $\mathcal{P}_k$, including the fixed point. The fixed point is therefore the coarsest stable refinement of the terminal partition. □

Proof of Theorem 4.2. Every exact edge $s \xrightarrow{a} s'$ induces the quotient edge $[s] \xrightarrow{a} [s']$, so projecting a path proves the forward direction. For the reverse direction, suppose a quotient path begins at $[s]$ with $[s] \xrightarrow{a_1} z_1$. Some representative $u \in [s]$ has an a_1-edge to a member of z_1. Stability says that s has an a_1-successor s_1 in the same class z_1. Repeating this argument lifts the full quotient path from the chosen representative s. Terminal type is constant within each class, so the lift reaches the same terminal type.

The two constructions preserve path length. Taking the minimum over all paths to a success terminal gives $d_{\mathrm{E}}(s) = d_{\mathrm{A}}([s])$, including infinity when neither path set is nonempty. Because the quotient map is surjective, the two graphs therefore have the same set of finite distance values and the same $d_{\max}$. Hence GraphGPO's replacement $d_{\max} + 1$ for unreachable nodes is also identical. For an

observed action a, stability gives the same set of successor classes, and all members of each successor class have the same distance by the preceding equality. Minimizing one plus successor distance therefore gives $Q_{\mathrm{E}}(s, a) = Q_{\mathrm{A}}([s], a)$. Substitution into Equation (4) proves raw-reward equality, and equality of all action costs preserves their ordering. □

Proof of Proposition 4.3. The exact sample mean is unbiased for μ_s and averages n_s independent noise terms, hence its variance and mean-squared error are σ^2/n_s. The pooled mean averages n_z independent noise terms and has expectation $\bar{\mu}_z$. Its variance is σ^2/n_z and its bias for μ_s is $\bar{\mu}_z - \mu_s$. The bias–variance decomposition gives Equation (19); comparing it with σ^2/n_s gives the stated condition. □

B QUOTIENT ABLATIONS

We ablate the two components of BQPO on WebShop with Qwen2.5-1.5B-Instruct, using the protocol of Section 5.

Always quotient. This variant keeps the behavioral quotient but removes the router: every transition uses the quotient reference group, i.e., $m_i = 1$ in Equation (16). Training remains stable. Across three independently trained runs, the variant reaches a success rate of $79.93 \pm 1.63\%$ and a task score of 89.97 ± 1.04 (mean and sample standard deviation), close to the routed means of $81.00 \pm 4.57\%$ and 90.30 ± 2.45 in Table 1. Quotient pooling already improves WebShop performance; the full method achieves the highest mean on both metrics.

Random quotient. This variant keeps the router unchanged but replaces the behavioral partition with a compression-matched random partition. For each rollout graph we compute the stable partition of Equation (6), then reassign exact nodes uniformly at random while preserving every class size. The random partition has the same class-size distribution and average group expansion, but can merge different terminal types and action-successor structures, so Theorem 4.2 no longer holds. Training collapses: the validation success rate stays near zero throughout, and the success rate of training rollouts remains below a few percent while the matched always-quotient run climbs above 50%. Group expansion alone is therefore insufficient: the behavioral structure of the quotient matters for successful pooling.