BEYOND GROUPING: ORTHOGONAL ADVANTAGE DECOMPOSITION FOR CRITIC-FREE LLM AGENT RL

Anonymous authors
Paper under double-blind review

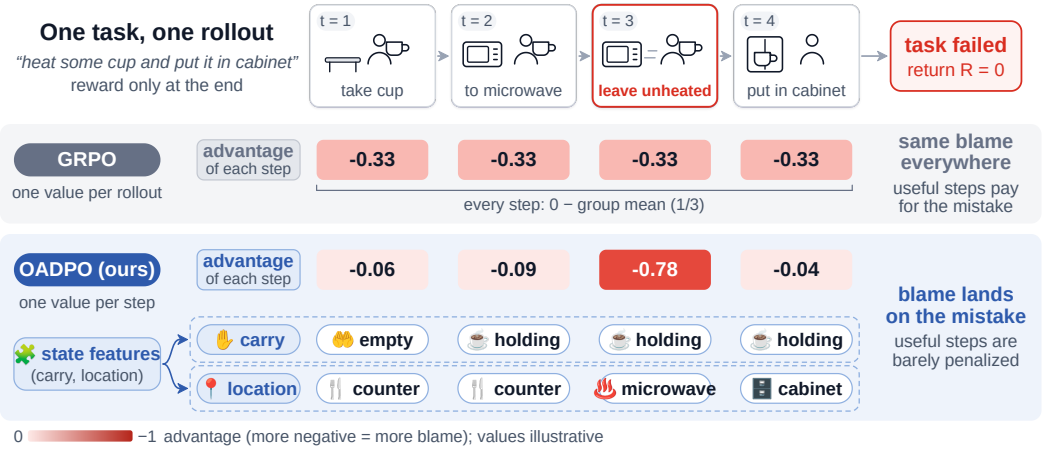

Figure 1: **The same failed rollout, credited two ways.** The agent loses the task at the step outlined in red (it leaves the microwave without heating the cup), so the rollout ends with $R = 0$. **GRPO** gives every step the same negative advantage. **OADPO** instead estimates a reference value per step from state features shared across rollouts. At the red step, the agent is holding the cup at the microwave, a state that is associated with a high expected return across other rollouts, so OADPO assigns this step a high reference value. Because the current rollout nevertheless fails, its realized return falls far below that reference, yielding a large negative advantage for the action taken at this state.

ABSTRACT

Group Relative Policy Optimization (GRPO) assigns the same rollout-level advantage to every step, which obscures credit assignment in multi-step LLM agents. Step-level methods such as GiGPO provide finer credit only when the same observation appears in multiple rollouts, but such exact matches are often sparse: in one ALFWorld reproduction, roughly one third of step groups had no match early in training. To move beyond exact grouping, we propose **Orthogonal Advantage Decomposition Policy Optimization** (OADPO), a critic-free method that replaces step grouping with a feature-based decomposition of expected return. OADPO shares evidence across nonidentical states through common feature components and assigns each component a closed-form shrinkage weight based on its estimated signal-to-noise ratio, allowing informative components to be trusted while noisy ones are suppressed. Because all steps in a rollout share the same outcome reward, OADPO corrects the noise estimate at the rollout level rather than treating steps as independent samples. Under independent features, the resulting weights are optimal within this estimator family and yield strictly lower prediction error than both GRPO's group mean and per-state averaging. The fitted state-dependent control variate is combined with a discounted step-level target, providing finer credit assignment without an additional critic, rollout, or model forward pass. Across ALFWorld, WebShop, and ScienceWorld with three model backbones, OADPO achieves the highest final success rate in seven of nine settings, including **96.4%**

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.

on ALFWorld with Llama-3.2-3B, while matching GRPO's per-step compute in a WebShop timing check. An ALFWorld ablation further shows that the fitted control variate contributes a 6.0-point gain beyond the discounted step target alone.

1 INTRODUCTION

LLMs are increasingly trained as agents that interact with environments over many steps, such as simulated households and web stores (Shridhar et al., 2021; Yao et al., 2022). These rollouts often span tens of steps but receive only a single outcome reward, positive upon task completion and 0 otherwise. Policy-gradient methods reduce gradient variance by subtracting a state-dependent control variate from the return, yielding an advantage that increases the probability of an action when positive and decreases it when negative (Williams, 1992; Greensmith et al., 2004). Actor-critic methods obtain this control variate from a learned value function (Mnih et al., 2016; Schulman et al., 2016), but for LLM agents the critic can be as large as the policy and difficult to train (Yuan et al., 2025b), motivating critic-free alternatives. GRPO (Shao et al., 2024) avoids a learned critic by sampling multiple rollouts for the same task and using their mean return as the control variate for every step.

Because GRPO uses a single group mean for the entire rollout, every step receives the same advantage. Figure 1 illustrates the problem on the ALFWorld task "heat some cup and put it in the cabinet." The agent picks up the cup and carries it to the microwave, then leaves without heating it and fails the task. GRPO nevertheless assigns the same negative advantage to the useful steps and the failure-causing action. Although these credit errors cancel on average, they add variance to each update, and this noise accumulates over longer rollouts.

Recent methods therefore refine credit at the step level. GiGPO (Feng et al., 2025) groups steps with identical observations and uses each group's mean return as their reference value. Exact matching can be too sparse, however: a step with no identical counterpart forms a singleton group, making its reference value equal to its own return and its step-level advantage zero. An independent ALFWorld reproduction found that such singleton groups accounted for 34.2% of step groups early in training and 20.7% late (Wang et al., 2026a). HGPO (He et al., 2026) mixes groups at multiple granularities with a tuned coefficient, while ProxMO and ProGPO (Fang et al., 2026; Fan & Liu, 2026) replace hard grouping with similarity-based weights defined by hand-designed kernels or constants. These methods therefore still rely on predefined notions of which steps should share a reference value.

Steps need not be identical to share useful information. For example, holding the cup at the microwave and holding it at the cabinet are different states, but both reveal that the agent still holds the target object. How often rollouts succeed from states sharing such features indicates how promising those states are and helps localize where a failed rollout likely went wrong. OADPO therefore represents each state using a small set of features and estimates its reference value by pooling returns across steps that share the corresponding feature values. It also models feature combinations, such as *holding the cup* while *at the microwave*, to capture information beyond either feature alone. These feature-based components form an orthogonal decomposition and can be weighted separately by their estimated signal-to-noise ratios, so components that capture meaningful variation across states receive more weight, while those dominated by estimation noise are shrunk toward zero. The resulting state-dependent control variate is then used in a discounted step-level term that provides a step-varying learning signal alongside GRPO's rollout-level advantage.

We call the method **Orthogonal Advantage Decomposition Policy Optimization** (OADPO). Its components are given by the functional ANOVA decomposition (Hoeffding, 1948) (Figure 2), which makes them orthogonal when the state features are independent. This orthogonality lets us weight each component separately, with an optimal closed-form shrinkage weight given by its signal relative to its signal plus noise (Wiener, 1949; James & Stein, 1961). OADPO estimates both quantities directly from each training batch, so the component weights require neither tuning nor learning. Reliable components, such as *carry* in Figure 2, retain most of their estimated effect, while noisy or sparsely observed components are shrunk toward zero. This turns the question of which steps should share a reference value into how much each shared component should be trusted. GRPO and per-state averaging are two limiting cases: GRPO uses only the overall mean and ignores state-dependent components, whereas per-state averaging uses all components at full strength. OADPO instead determines how much to trust each component from its estimated signal and noise. Under independent features, we prove that these optimal weights achieve strictly lower prediction error

than both limiting cases whenever at least one component contains both signal and estimation noise (Theorem 1 and Corollary 2); with the approximately independent features used in practice, this guarantee becomes approximate.

Because each rollout provides only one outcome reward, its many steps do not constitute independent reward observations. This has two consequences for OADPO. First, treating steps as independent would underestimate estimation noise, so OADPO corrects the noise at the rollout level (Section 4.4). Second, the number of rollouts, rather than the number of steps, limits how many feature effects can be reliably identified (Theorem 3). OADPO therefore selects a small set of features during a short GRPO warm-up and keeps them fixed for the rest of training.

We compare OADPO with GRPO, GiGPO, and HGPO (Shao et al., 2024; Feng et al., 2025; He et al., 2026) on ALFWorld, WebShop, and ScienceWorld (Shridhar et al., 2021; Yao et al., 2022; Wang et al., 2022), using three model backbones under a shared codebase and matched training budgets. Our contributions are:

- **A critic-free, state-dependent control variate for step-level credit assignment.** OADPO represents each state with a set of features and decomposes its expected return into orthogonal components that can share evidence across nonidentical states. This turns the question of which steps should be grouped into how much each shared component should be trusted.
- **Closed-form component weights with theoretical guarantees.** We derive a closed-form shrinkage weight for each component from its estimated signal and rollout-corrected noise. Under independent features, these weights minimize prediction error within the estimator family and achieve strictly lower error than GRPO's group mean and per-state averaging. We also establish an identifiability limit showing that outcome-only rewards provide at most rollout-level information for estimating feature effects, motivating a compact feature set.
- **A controlled evaluation across environments and backbones.** OADPO achieves the highest final success rate in seven of nine settings, including 96.4% on ALFWorld with Llama-3.2-3B, compared with 91.7% for the next highest method. In a WebShop timing check, its per-step compute matches GRPO's, and ALFWorld ablations show that both the fitted control variate and the discounted step target contribute to the improvement.

2 Related Work

Critic-free group-relative methods for LLM agents. GRPO (Shao et al., 2024) replaces the critic with the mean return of G rollouts of the same prompt, a Monte-Carlo control variate in the line of REINFORCE and RLOO (Williams, 1992; Ahmadian et al., 2024). Its agent successors keep this template and change which steps share a reference value. Tree-GRPO groups steps by tree node (Ji et al., 2026), BiPACE clusters hidden states within a tunable radius (Wang et al., 2026a), GiGPO groups steps with identical observations (Feng et al., 2025), HGPO mixes several group levels with a tuned coefficient (He et al., 2026), and GEPO and G^{2}PO share credit over a state-transition graph (Yuan et al., 2025a; Wang et al., 2026c). Other multi-turn methods give every turn the rollout-level advantage (Li et al., 2026) or truncate a rollout once the agent's beliefs drift (Zou et al., 2026). All of them fix in advance which steps are comparable. OADPO lets every step contribute to shared parts of the control variate and estimates from the data how much each part counts (details in Appendix A).

Soft grouping. ProxMO (Fang et al., 2026) and ProGPO (Fan & Liu, 2026) replace hard groups with weighted mixtures, using a TF-IDF similarity kernel and inverse-variance weights respectively, and EBPO (Han et al., 2026) shrinks each prompt's group mean toward the batch mean with a single signal-to-noise ratio. Their weights come from a hand-chosen kernel, tuned constants or one scalar; OADPO derives one closed-form weight per orthogonal part, zero for a part without signal.

Control variates in policy gradients. Subtracting a state-dependent reference value to reduce the variance of the policy gradient is classical (Williams, 1992; Greensmith et al., 2004; Weaver & Tao, 2001). Action-dependent variants derive an optimal reference per factor of a factorized policy (Wu et al., 2018), though Tucker et al. (2018) question their gain over state-dependent ones, and Q-Prop uses a learned critic as a control variate (Gu et al., 2017); implementation details can matter as much as the estimator (Engstrom et al., 2020). These methods need a critic or a factorized action. OADPO stays critic-free and factorizes the *state*, shrinking each part by its signal-to-noise ratio.

Shrinkage, functional ANOVA and clustered samples. The part weights of OADPO follow the James–Stein (Wiener) shrinkage rule (James & Stein, 1961; Efron & Morris, 1973), the parts are the functional ANOVA (Hoeffding) decomposition (Hoeffding, 1948; Sobol', 2001), and the noise correction is Kish's design effect for clustered samples (Kish, 1965). Each tool is used as published; their combination is specific to rollout data: orthogonality makes shrinkage separable across many parts, and the design effect keeps the noise estimate valid when a rollout's steps share one reward.

3 PRELIMINARIES

Setting. An LLM policy π_θ acts in a text environment. For each task (prompt) u we sample G rollouts. Rollout τ visits states $s_1, \dots, s_{T_\tau}$ and succeeds or fails. Following verl-agent, all three environments pay a success reward of 10 and 0 otherwise, and every step whose response fails the output-format check (no well-formed `<action>... </action>` block) is charged an invalid-action penalty of 0.1. The reward credited to step i of rollout τ is therefore

$$R_i \;=\; 10 \cdot \mathbf{1}[\tau \text{ succeeds}] \;-\; 0.1 \cdot v_i, \qquad v_i \in \{0, 1\}, \tag{1}$$

where $v_i = 1$ marks an invalid action. Every estimator in this paper, GRPO's included, is computed from R_i; we write R_τ for its outcome part $10 \cdot \mathbf{1}[\tau \text{ succeeds}]$, which all steps of a rollout share, and drop the step index when no confusion arises. A batch holds n_τ rollouts of n_u prompts and stores one record (state, action, reward) per step i.

Control variates and advantages. The policy gradient with a state-dependent control variate b is $\nabla_\theta J = \mathbb{E}_\pi[\sum_t (R - b(s_t)) \nabla_\theta \log \pi_\theta(a_t \mid s_t)]$, and $A(s_t, a_t) = R - b(s_t)$ is the usual advantage of action a_t. The gradient is unbiased for *any* b that depends on the state s_t alone, because $\mathbb{E}_{a \sim \pi}[\nabla_\theta \log \pi_\theta(a \mid s)] = 0$; the control variate changes only the variance, and a good one makes each update much less noisy (Williams, 1992; Greensmith et al., 2004). The natural target for b is the expected return from the state, $\mathbb{E}[R \mid s]$. Critic-free group methods estimate it by averaging the returns of "comparable" steps,

$$\mathcal{G}(s) = \{(s', a', R') : s' \sim s\}, \quad b(s) = \text{mean}\{R' : (s', a', R') \in \mathcal{G}(s)\}, \quad A(s, a) = R - b(s), \tag{2}$$

and differ only in the relation $\sim$ that defines "comparable": GRPO uses "same prompt" and GiGPO "same observation" (published choices: Table 3).

Features and targets. We describe a state by K discrete features $x = (x_1, \dots, x_K)$; feature x_k takes c_k values (Section 4.6). The control variate predicts a target y_i, either the step reward, $y_i = R_i$, or the discounted return-to-go, $y_i = y_i^{\text{RTG}} = \gamma^{T_\tau - t_i} R_i$. The discount γ enters only OADPO's step term (Section 4.7), as the regression target and as the step-level return; the discount of PPO, the policy-update rule we use, stays at 1.0. The control variate estimates $f(s) = \mathbb{E}[y \mid s]$. In OADPO the overall mean of f is taken to be the group mean $\bar{y}_g$ that GRPO already subtracts (Section 4.7), so the fitted part $\hat{b}$ models the group-centred residual $y - \bar{y}_g$ and the weights of Section 4.3 act on the parts beyond the overall mean. We write ν for the distribution of states that the current policy visits, so that $f \in L^2(\nu)$, and $\sigma^2 = \text{Var}(y \mid s)$ for the noise around f.

4 ORTHOGONAL ADVANTAGE DECOMPOSITION POLICY OPTIMIZATION

4.1 MOTIVATION AND OVERVIEW

OADPO gives every step its own control variate, an estimate of how well rollouts usually end from that step's state, without a critic and without fixing in advance which steps are comparable. Grouping, as in GiGPO, is all-or-nothing: two steps either share a reference value or ignore each other, and HGPO only mixes several such groupings with a tuned coefficient. OADPO shares partially: it writes a step's expected return as a sum of parts that each depend on only some features. For example, the part for "does the agent hold the target object?" draws on every step in the batch, so even a step with a rare observation gets an informed reference value. In Figure 2 (right), rollouts that reach the microwave holding the cup usually succeed, so when the rollout of Figure 1 fails anyway the blame concentrates on that step. Proofs are deferred to Appendix B.

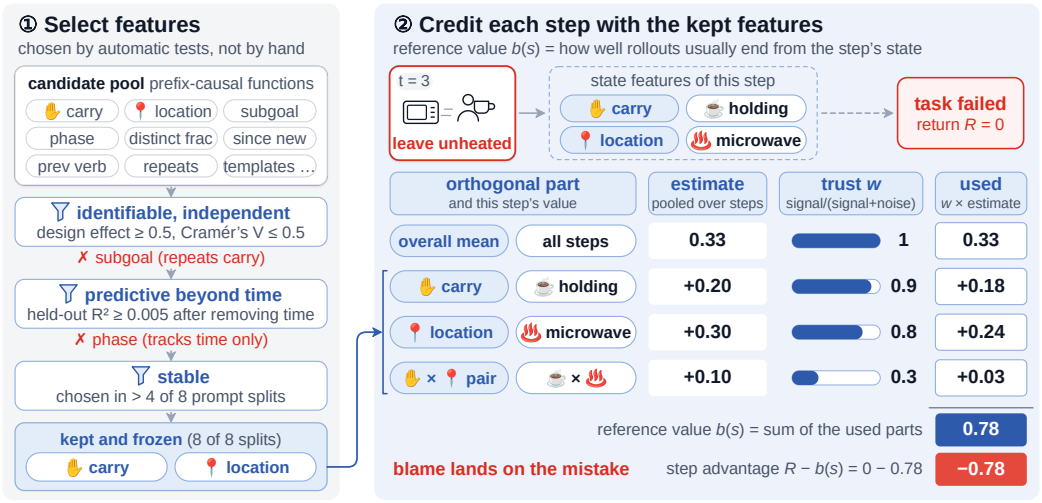

Figure 2: **How OADPO chooses its features and credits one step.** *Left:* automatic tests, not the authors, pick the features at the end of warm-up (Section 4.6); on ALFWorld, under the current gates, *carry* and *location* survive. *Right:* for the failed step of Figure 1, each orthogonal part is estimated from all steps that share its feature values and used in proportion to its trust $w_U = \sigma_U^2/(\sigma_U^2 + \eta_U)$ (Sections 4.3 and 4.4); the sum $b(s)$ is high, so $R - b(s)$ blames this step (Equation (4)). Rewards on a 1/0 scale; values illustrative.

4.2 Defining the parts: an orthogonal decomposition

Setting each part's weight on its own requires parts that do not overlap. If two parts explained the same variation, as a coarse group and a finer group nested inside it do, the error of one would depend on the weight of the other, and the two weights would need joint tuning.

The functional ANOVA (Hoeffding) decomposition (Hoeffding, 1948; Sobol', 2001) gives non-overlapping parts. For every subset U of the features it defines a part $f_U(x_U)$ that depends only on the features in U. The part $f_\emptyset$ is the overall mean; $f_{\{k\}}$ is the main effect of feature k, how much the expected return changes with x_k on average; and $f_{\{k,l\}}$ is the interaction of x_k and x_l, what the pair adds beyond the two main effects. The parts sum to f, and they are mutually orthogonal under the following assumption (Theorem 5).

Assumption 1 (Independent features)**.** Under the behaviour policy, the K state features are independent: $\nu = \bigotimes_{k=1}^{K} \nu_k$.

The assumption holds only approximately, and a violation cannot bias learning, because the gradient is unbiased for *any* control variate that depends on the state alone (up to a small in-sample effect of fitting $\hat{b}$ on the current batch; Appendix C); it only makes the error split of Section 4.3 approximate, so the weights are no longer exactly optimal (Appendix B.6). Feature selection keeps the assumption close to true by dropping any feature strongly associated with one already kept (Section 4.6).

The decomposition by itself changes nothing: adding all parts at full strength gives back the mean return of each combination of feature values. The control variate changes only through how strongly it uses each part. We therefore consider the family

$$b_w(s) = \sum\nolimits_U w_U \, \hat{f}_U(x_U), \qquad w_U \in [0, 1],$$

where $\hat{f}_U$ is the batch estimate of part U. Existing methods sit at the corners of this family. GRPO keeps only the overall mean ($w_\emptyset = 1$, all other $w_U = 0$), and the saturated estimator that averages the returns within each combination of feature values, the step-level component of GiGPO and what we refer to as per-state averaging, keeps every part at full strength ($w_U = 1$ for all U; with the interaction order truncated at 2, as deployed, "every part" means every part up to that order).

4.3 TRUSTING EACH PART: CLOSED-FORM WEIGHTS

A part estimated from few or noisy steps adds more noise than signal: per-state averaging uses it at full strength and makes the control variate worse, while GRPO drops every part and loses the signal with the noise. The best weight lies between these extremes and differs from part to part.

Let $\sigma_U^2 = \|f_U\|^2$ be the signal energy of part U and η_U the mean-squared error of its estimate $\hat{f}_U$. We assume unbiased estimates with errors uncorrelated across parts and independent of f (Assumption 2).

Theorem 1 (Optimal weights)**.** *Under Assumptions 1 and 2, the error of the control variate splits into one term per part,* $\mathbb{E}\|b_w - f\|^2 = \sum_U [(1-w_U)^2\sigma_U^2 + w_U^2\eta_U]$*, and is minimized by*

$$w_U^\star = \frac{\sigma_U^2}{\sigma_U^2 + \eta_U} \in [0,1], \qquad \mathrm{MSE}^\star = \sum_U \frac{\sigma_U^2\,\eta_U}{\sigma_U^2 + \eta_U}. \tag{3}$$

Corollary 2 (Better than both corners)**.** $\mathrm{MSE}^\star < \min(\mathrm{MSE}_{\mathrm{GRPO}}, \mathrm{MSE}_{\mathrm{sat}})$*, where* sat *denotes per-state (saturated) averaging,* $w_U \equiv 1$*, and* $\mathrm{MSE}_{\mathrm{GRPO}} = \eta_\emptyset + \sum_{U\neq\emptyset}\sigma_U^2$ *and* $\mathrm{MSE}_{\mathrm{sat}} = \sum_U \eta_U$*, as soon as one part* $U \neq \emptyset$ *has both* $\sigma_U^2 > 0$ *and* $\eta_U > 0$*.*

The optimal weight is the classical Wiener (James–Stein) shrinkage factor (Wiener, 1949; James & Stein, 1961), the part's signal as a share of its signal plus noise: a part with no signal gets weight 0, and a part measured without noise gets weight 1. The overall-mean part is kept at weight 1 by construction, since it is GRPO's group mean (Section 3); shrinkage acts on the parts beyond it. The error splits into independent terms because the parts are orthogonal and their estimation errors are uncorrelated, which is what gives a closed form; with nested groups, as in HGPO, the terms interact and a mixing coefficient needs tuning. In practice σ_U^2 is unknown. OADPO estimates it from the batch with the positive-part James–Stein rule (Efron & Morris, 1973) (Equation (8)) and smooths each part's fitted energy $\|\hat{f}_U\|^2/N$ and noise term η_U across training steps with an exponential moving average (EMA, momentum 0.9; Algorithm 1). Corollary 2 holds for the optimal weights, and the experiments test whether the estimated weights realize the gain.

4.4 MEASURING THE NOISE WHEN STEPS SHARE A REWARD

The textbook estimate of η_U in Equation (3), $d_U\sigma^2/N$ for a part with d_U free parameters and N samples, assumes independent samples. Agent data break it: all T steps of a rollout carry the same final reward, so a feature that stays constant within a rollout gains about one independent observation per rollout, not one per step, and the textbook estimate mistakes noise for signal.

OADPO multiplies the textbook estimate by a design effect DE_U (Kish, 1965). It sums each part's contributions within a rollout and measures the spread of these sums across rollouts:

$$\eta_U = \mathrm{DE}_U \cdot \frac{d_U\,\sigma^2}{N} \qquad \text{(exact definition in Equation (9), Appendix B.7).}$$

$\mathrm{DE}_U = 1$ recovers independent samples, and $\mathrm{DE}_U \approx T$ corresponds to a feature that is constant within a rollout. The correction must use the same target as the regression: during development, a mismatch gave one feature a design effect of 0.19 instead of 9.31 (Appendix B.7).

4.5 OUTCOME REWARDS LIMIT WHAT THE DATA CAN IDENTIFY

With one reward per rollout, more steps from the same rollout add no new reward information, which limits how many parameters the data can identify (proof: Theorem 6).

Theorem 3 (Identifiability limit)**.** *Let the target be any deterministic reweighting of the outcome reward,* $y_i = h(t_i, T_{g(i)})R_{g(i)}$ *(ignoring the* 0.1 *penalty of Equation* (1)*), where* $g(i)$ *is the rollout that contains step* i*. A step-level regression depends on the rewards only through* n_τ *rollout-level sums, so the data identify at most* $\min(p, n_\tau)$ *of its* p *directions, however many steps there are.*

Corollary 4. (i) *A part is unidentifiable exactly when its rollout-level sums vanish, which is also the case* $\mathrm{DE}_U = 0$*.* (ii) *Such a part has* $\hat{\sigma}_U^2 = 0$ *and weight* 0*.* (iii) *The effective sample size is* $N_{\mathrm{eff}} = n_\tau - n_u$*, not the number of steps.*

On ALFWorld, $19{,}473$ step records logged before RL (Section 5.3) give only $N_{\mathrm{eff}} = 308$.

4.6 CHOOSING THE FEATURES

Theorem 3 leaves room for only a few parts, so OADPO chooses the features with care.

Candidate features are cheap, hand-specified functions of the trajectory prefix (the observations up to the current step and the actions before it); they never use the current action, future steps or the reward (Appendix D). Which of them OADPO uses is decided during warm-up by automatic tests, not by hand (Figure 2, left): a feature is kept only if it meets three requirements. It must be *identifiable*: at most 6 bins and a design effect of at least 0.5. It must be *approximately independent* of the features already kept: Cramér's $V \leq 0.5$. And it must be *predictive beyond elapsed time* on held-out prompts: $R^2 \geq 0.005$, a floor motivated by $w_U^\star > 1/2$, which requires $R_U^2 > \mathrm{rank}_U \, \mathrm{DE}_U / N$ and is of this order at our sample sizes (Appendix B.8). Without the last check, "time until the rollout ends" passes for state information. Selection runs on several prompt splits and keeps the features that a majority of splits choose; the feature set and the interaction order then stay frozen, while the feature bins are refreshed every 40 steps (Appendix B.8, Table 4). One deployment detail is disclosed in Appendix C.5: the ALFWorld run used a specification frozen under an earlier gate ordering, which retains one feature the current pipeline rejects as a step-index proxy.

4.7 THE ADVANTAGE USED IN TRAINING

Under the reward of Equation (1), every step of a rollout receives the same value apart from the 0.1 invalid-action penalty, so a step-level term needs a target that varies within the rollout. OADPO uses the discounted return-to-go $y^{\mathrm{RTG}} = \gamma^{T-t} R$, which is larger for steps closer to a success. This target also rewards finishing sooner: in our ALFWorld data, rollout length and success are strongly anti-correlated ($\mathrm{corr}(T, R) = -0.686$). The outcome reward should keep driving learning, and OADPO should reduce exactly to GRPO when it has nothing to add, so OADPO keeps GRPO's advantage unchanged and adds a step-level term that carries the discounted target and the fitted control variate:

$$A \;=\; \underbrace{\frac{R - \bar{R}_g}{\mathrm{std}_g}}_{A^{\mathrm{ep}} \,=\, \text{GRPO}} \;+\; w_{\mathrm{step}} \cdot \underbrace{\frac{y^{\mathrm{RTG}} - \bar{y}_g - \hat{b}(s_t)}{\mathrm{std}_g^{\mathrm{RTG}}}}_{A^{\mathrm{step}} \,=\, \text{step term of OADPO}} \;. \tag{4}$$

The rollout term A^{ep} contains no fitted quantity and equals GRPO's advantage; setting $w_{\mathrm{step}} = 0$ recovers GRPO exactly on the de-duplicated step batch, and if selection keeps no feature, OADPO drops the step term (Algorithm 1). The step term A^{step} affects learning in two ways: its discounted target changes what the policy is rewarded for, and the fitted control variate $\hat{b}$ changes only the variance; Section 5.2 separates the two. We fix the scalar w_{step} at 1.0 and never tune it, like GiGPO's step-level coefficient; every part weight is closed-form. The fit costs one thin SVD per part per training step, with no extra rollout, model or forward pass (Appendices B.9 and C).

5 EXPERIMENTS

Setup. All runs use verl-agent (Feng, 2025; Sheng et al., 2025) with $G = 8$ rollouts per task, 16 tasks (128 rollouts) per optimization step and 2 to 4 B200 GPUs per job. The Llama-3.2-3B-Instruct (Grattafiori et al., 2024) runs train for 300 steps on ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022); ScienceWorld (Wang et al., 2022) is described in Section 5.4. ALFWorld arms evaluate on `valid_seen` every 10 steps; WebShop arms start from a behaviour-cloning checkpoint and evaluate on all 500 test goals every 20 steps. The Llama-3.1-8B and SmolLM3-3B (Bakouch et al., 2025) runs follow Appendix E.

Baselines and comparability. We compare four advantage estimators, GRPO, GiGPO, HGPO and OADPO, and re-train every baseline under one wrapper and one protocol per column of Table 1. Unless marked as seed 0, every success rate is the mean over three seeds at the last evaluation point of the run, as in the GiGPO and HGPO papers.

Table 1: Success rate (%) on ALFWorld (ALF), WebShop (Web) and ScienceWorld (Sci) at the final evaluation point, mean over three seeds; ScienceWorld is evaluated on the full dev set (Appendix E). Bold and underlined: best in column.

	Llama-3.2-3B-Instruct			Llama-3.1-8B-Instruct			SmolLM3-3B		
Method	ALF	Web	Sci	ALF	Web	Sci	ALF	Web	Sci
GRPO	59.6	78.9	17.5	92.9	76.6	24.2	84.3	76.6	37.5
GiGPO	91.7	**81.7**	49.6	96.4	81.2	37.5	93.6	78.0	46.3
HGPO	90.9	79.9	52.5	95.0	77.8	47.1	93.6	79.0	**50.4**
OADPO (ours)	**96.4**	80.9	**56.7**	**97.1**	**81.8**	**47.9**	**95.0**	**79.8**	49.9

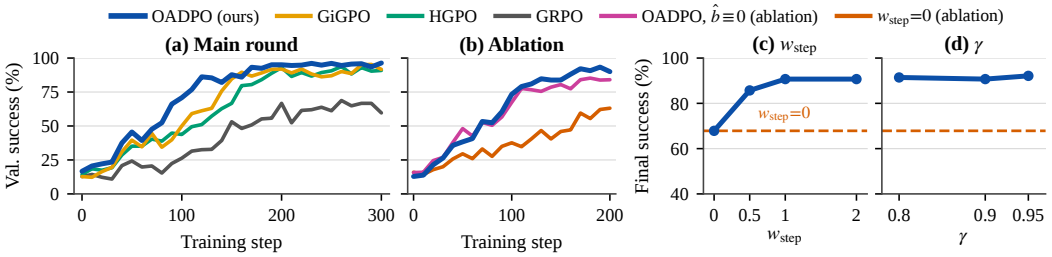

Figure 3: ALFWorld, Llama-3.2-3B. (a) Validation success over training, mean over three seeds; the reported number is the last point (other backbones: Table 11). (b) Ablation (200 steps, three seeds; Table 2). (c, d) Step-200 success against w_{step} and γ (seed 0; dashed line: $w_{\text{step}} = 0$).

5.1 Main results

OADPO reaches the highest final success on ALFWorld and reaches it first. With Llama-3.2-3B, OADPO ends at 96.4%, 4.7 points above GiGPO (91.7%), 5.5 above HGPO (90.9%) and 36.8 above GRPO (59.6%), and it is the first arm to cross 90% validation success (Figure 3a, Table 10). On WebShop the four arms end within 2.8 points of each other, with GiGPO highest.

The ranking transfers to a larger model and to a different model family. With Llama-3.1-8B and with SmolLM3-3B, OADPO again ends highest on ALFWorld and, unlike at 3B, also on WebShop, by 0.6 to 1.4 points over the runner-up (Table 11); the margins are smaller at 8B, where GRPO itself reaches 92.9% on ALFWorld under a different protocol (Appendix E). On ScienceWorld, OADPO ends highest with both Llama backbones and 0.5 points below HGPO with SmolLM3-3B (Section 5.4). Across the nine settings, OADPO's worst rank is second, by at most 0.8 points.

5.2 Both the fitted control variate and the discounted target contribute

Both ingredients carry the gain. The step term of Equation (4) has two ingredients, the fitted control variate $\hat{b}$ and the γ^{T-t}-discounted target; Table 2 (top) separates them with three arms of three seeds each (Figure 3b). Removing the step term costs 26.9 points and lowers success on every seed; the w_{step}=0 arm and GRPO compute the same estimator on the de-duplicated batch (Appendix C). Zeroing the fitted control variate costs 6.0 points (per seed -4.3, -7.2 and -6.4, all same sign).

The fitted control variate carries its own share. The $\hat{b} \equiv 0$ arm still fits the control variate and only stops using it (its fit explains 0.29 of the step-target variance at the end, against 0.27 in the full arm), so the step term's 26.9 points split into 20.9 from the discounted target, which favours rollouts that succeed in fewer steps, and 6.0 from the fitted control variate.

Sensitivity to the design choices. On seed 0, four changes reach the 5-point threshold: $w_{\text{step}} = 0$ (-22.8 points; Table 13), and in Table 2 (bottom) removing the second-order parts (-9.3), the `carry`-only feature set (-5.7) and $w_{\text{step}} = 0.5$ (-5.0). Success saturates at $w_{\text{step}} = 1$ (Figure 3c), $\gamma \in [0.8, 0.95]$ moves it by at most 1.4 points (Figure 3d), and the remaining changes stay below the

Table 2: Ablation on ALFWorld (Llama-3.2-3B, 200 steps, final point, success %). Top: mean of three seeds, Δ paired within seed (per seed: Table 13). Bottom: one change at a time, seed 0, Δ against the seed-0 full arm (90.7%); single-seed differences below 5 points are not interpreted (re-binning, EMA: Table 4).

	Variant	Success (%)	Δ (points)
3 seeds	OADPO (γ=0.9, w_{step}=1, fitted $\hat{b}$)	**90.0**	ref.
	fitted control variate off ($\hat{b} \equiv 0$; discounted target only)	84.0	-6.0
	step term off (w_{step}=0; $\equiv$ GRPO)	63.1	-26.9
seed 0	$\gamma = 0.8/0.95$ (deployed 0.9)	91.4/92.1	$+0.7/+1.4$
	$w_{\text{step}} = 0.5/2$ (deployed 1)	85.7/90.7	$-5.0/\ \ 0.0$
	features `carry`/{`carry`,`loc`} (deployed: 3)	85.0/92.9	$-5.7/+2.2$
	no 2nd order/no re-binning/EMA 0.99	81.4/86.4/87.9	$-9.3/-4.3/-2.8$
	single-term advantage (A^{step} without A^{ep})	92.1	$+1.4$

threshold. The single-term advantage ends similarly (92.1% vs. 90.7%) but reaches 60% at step 150 instead of 90: 70% of its early records hit the standard-deviation floor (32% with both terms).

5.3 Cost, engagement and the offline signal

On WebShop, OADPO costs the same as GRPO. OADPO and GRPO take 64.5 and 64.6 seconds per step on WebShop (Table 10), so the cost of fitting the control variate is below the timer's resolution. GiGPO and HGPO cost 16% and 36% more, because their response lengths jump mid-training ($\times$3.5 and $\times$4.4; Figure 5, Appendix G), not because of their advantage computation.

Offline R^2 checks whether the features carry signal. Before RL, running the selection stage on logged rollouts gives a held-out R^2 of $\mathbf{0.1118 \pm 0.0084}$ on ALFWorld and $\mathbf{0.0678 \pm 0.0100}$ on WebShop at $\gamma = 0.9$ (Table 9). During training, every reported run passes an engagement checklist at every step after warm-up, and the fit cuts GRPO's prediction error by a median of 17% on ALFWorld and 37% on WebShop (Table 12).

5.4 ScienceWorld: the signal lies in feature interactions

On ScienceWorld the standard selection keeps no feature (Appendix F), because 82% of the variance the features explain beyond task and time lies in interaction layers (Table 7), which a one-feature-at-a-time screen cannot detect. Selecting interaction parts jointly, with the same checks and budget, recovers a feature set in every prompt split (detrended $R^2 = \mathbf{0.1250 \pm 0.0097}$). With these features OADPO ends highest with both Llama backbones ($+4.2$ and $+0.8$ points over HGPO) and 0.5 points below HGPO with SmolLM3-3B (Tables 8 and 11, Figure 4). ScienceWorld has no $\hat{b} \equiv 0$ arm to separate the two ingredients.

6 Conclusion and Limitations

Critic-free RL for LLM agents has mostly asked which steps to group. OADPO asks how much to trust each orthogonal part of a step's expected return, answers in closed form, and reaches the highest final success rate in seven of nine settings, at GRPO's per-step cost in a timing check on WebShop. Two limitations matter most (Appendix G.6). **First**, the method depends on the features: when they miss the states that matter for the outcome, no part beyond the overall mean carries signal and OADPO reduces to GRPO; ScienceWorld shows a milder form, with the signal hidden in feature interactions (Section 5.4). **Second**, the candidate features are hand-specified per environment (Appendix D); selection among them is automatic, but the pool is not, and learning features from the observation text remains future work.

AI USE STATEMENT

In this work, we used generative AI tools for four purposes. To aid or polish writing: editing the text for readability and formatting references. For retrieval and discovery: identifying related work and assisting with translation of non-English source material consulted during the literature search (e.g., in French and Chinese) into English. To draft sections of the paper: drafting parts of the text. To implement the method and its supporting code: parts of the source code of the estimator and its documentation, LaTeX tables, figure code (including the diagrams in Figures 1 and 2) and analysis and plotting scripts were written with AI assistance. We have not used generative AI tools for research ideation (developing theoretical models, proposing hypotheses, designing the method or the experiments, or interpreting results), for generating synthetic datasets, or for formulating or proving mathematical claims; cleaning or reformatting datasets and qualitative or thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work: the authors read and verified every reference found with AI assistance, recomputed every numerical value in the tables and figures from the raw per-step logs, and read, tested and verified the correctness of all AI-assisted code, including the implementation with which every reported experiment was run. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work trains and evaluates language-model agents in three simulated text environments (household tasks, a mock shopping site and a science laboratory simulator). No human subjects are involved; we collect or release no personal data and introduce no new dataset. The policies are open-weight models (Llama-3.2-3B, Llama-3.1-8B, SmolLM3-3B), and all environments are established public benchmarks used under their original licences. As an advantage estimator, the method shares the general risk of reinforcement learning, that training amplifies whatever the reward function measures, and does not add an application-specific concern beyond it.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix B specify the estimator completely, and Appendix C gives pseudocode for the online loop (Algorithm 1), the fit (Algorithm 2) and the selection stage (Algorithm 3). Appendix C also lists every hyperparameter of OADPO, including the thresholds of the feature checks, and reproduces the frozen ALFWorld and WebShop specifications (feature sets, bin counts, observed marginals and, for WebShop, cut points). Every theoretical claim states its assumptions (Assumptions 1 and 2); Appendix B proves each claim and describes what degrades when Assumption 1 fails. Appendix E gives the experimental protocol: split sizes, evaluation pools and frequencies. The source code of OADPO (the estimator, the feature construction and selection stages, the frozen feature specifications, and the training and evaluation entry points) is provided as a compressed archive in the supplementary material.

REFERENCES

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

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 12461–12495, 2024.

Elie Bakouch, Carlos Miguel Patiño, Anton Lozhkov, Edward Beeching, Aymeric Roucher, Nouamane Tazi, Aksel Joonas Reedi, Guilherme Penedo, Hynek Kydlicek, Clémentine Fourrier, Nathan Habib, Kashif Rasul, Quentin Gallouédec, Hugo Larcher, Mathieu Morlon, Joshua Lochner, Vaibhav Srivastav, Xuan-Son Nguyen, Colin Raffel, Lewis Tunstall, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. Hugging Face blog, `https://huggingface.co/blog/smollm3`, 2025.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3676–3713, 2023.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Sanjiban Choudhury. Process reward models for LLM agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*, 2025.

Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

Bradley Efron and Carl Morris. Stein's estimation rule and its competitors—an empirical Bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020.

Mingxuan Fan and Peiyang Liu. Progress- and reliability-oriented group policy optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.04242*, 2026.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, and Chang Liu. Proximity-based multi-turn optimization: Practical credit assignment for LLM agent training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 285–307, 2026.

Lang Feng. verl-agent. GitHub repository, `https://github.com/langfengQ/verl-agent`, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46375–46408, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

Shixiang Gu, Timothy P. Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. Q-Prop: Sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations*, 2017.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in RL for large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Kevin Han, Yuhang Zhou, Mingze Gao, Gedi Zhou, Serena Li, Abhishek Kumar, Xiangjun Fan, Weiwei Li, and Lizhu Zhang. EBPO: Empirical Bayes shrinkage for stabilizing group-relative policy optimization. *arXiv preprint arXiv:2602.05165*, 2026.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations*, 2026.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

Willard James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 361–379, 1961.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *International Conference on Learning Representations*, 2026.

Leslie Kish. *Survey Sampling*. John Wiley & Sons, 1965.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. In-the-flow agentic system optimization for effective planning and tool use. In *International Conference on Learning Representations*, 2026.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Jianbin Jiao, and Junge Zhang. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations*, 2026.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, 2016.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. HiPER: Hierarchical plan–execute RL for multi-turn LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Leyang Shen, Yang Zhang, Chun Kai Ling, Xiaoyan Zhao, and Tat-Seng Chua. CARL: Criticality-aware agentic reinforcement learning. *arXiv preprint arXiv:2512.04949*, 2025.

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 *International Conference on Learning Representations*, 2021.

Ilya M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1–3):271–280, 2001.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7584–7600, 2024.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E. Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5015–5024, 2018.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. SPA-RL: Reinforcing LLM agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025a.

Hanyang Wang, Weijieying Ren, Yuxiang Zhang, Ding Cao, Zhizhao Zeng, Ke Zeng, and Tianxiang Zhao. BiPACE: Bisimulation-guided policy optimization with action counterfactual estimation for LLM agents. *arXiv preprint arXiv:2606.25556*, 2026a.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Xucong Wang, Ziyu Ma, Yong Wang, Yuxiang Ji, Shidong Yang, Guanhua Chen, Pengkun Wang, and Xiangxiang Chu. APPO: Agentic procedural policy optimization. *arXiv preprint arXiv:2606.12384*, 2026b.

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

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 538–545, 2001.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via fine-grained reward structure and credit assignment. *arXiv preprint arXiv:2505.11821*, 2025.

Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications*. The Technology Press of the Massachusetts Institute of Technology and John Wiley & Sons, 1949.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961, 2025.

Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. Progress-conditioned group policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2607.22724*, 2026.

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*, volume 35, pp. 20744–20757, 2022.

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agent-Tuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209, 2024.

Siyuan Zhu, Chao Yu, Rongxin Yang, Zongkai Liu, Jinjun Hu, Qiwen Chen, and Yibo Zhang. GAGPO: Generalized advantage grouped policy optimization. *arXiv preprint arXiv:2605.13217*, 2026.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Garry Yang, Mufei Li, Qing Da, James Cheng, Pan Li, and Yu Gong. Reducing belief deviation in reinforcement learning for active reasoning of LLM agents. In *International Conference on Learning Representations*, 2026.

A EXTENDED RELATED WORK

Section 2 is compressed for length; Table 3 lists the group-forming relation and the free parameters of each critic-free method we compare with, and the paragraphs below position OADPO against the closest prior methods.

Table 3: Critic-free group-relative methods for LLM agents, organized by how they form the baseline group. The first block fixes a single equivalence relation; the second softens it into a weighted or graph-structured mixture at the cost of free parameters. OADPO abandons the partition entirely. Free parameters are the method-specific ones named in each paper (ProxMO: softmax temperature and modulation strength β; ProGPO: history depth, neighbour count, two credit scales, a variance floor and a similarity temperature), excluding the PPO clip range and KL coefficient every method shares; OADPO's remaining global scalar w_{step} and its selection thresholds are listed in Table 4.

Method	Relation $s' \sim s$	Granularity	Free parameter
GRPO (Shao et al., 2024)	same prompt	coarsest	none
Tree-GRPO (Ji et al., 2026)	same tree node	medium	branching budget
GiGPO (Feng et al., 2025)	exact observation hash	finest	step/episode mix
BiPACE (Wang et al., 2026a)	hidden-state cosine cluster	tunable	ε
HGPO (He et al., 2026)	nested chain of the above	mixture	mixing scalar α
ProxMO (Fang et al., 2026)	TF-IDF-cosine softmax	soft	temperature, β
ProGPO (Fan & Liu, 2026)	inverse-variance mixture	mixture	≥ 6 constants
GEPO (Yuan et al., 2025a)	state-transition graph	graph node	criticality weights
G^{2}PO (Wang et al., 2026c)	shared-observation graph	graph node	graph-wide TD scaling
OADPO (ours)	*none*: orthogonal part weights	continuous	*closed form*

Group-relative optimization for multi-turn agents. GRPO was first adapted to interactive environments by treating the whole trajectory as the response (Wang et al., 2025b; Xi et al., 2025); earlier agent RL used PPO with a critic (Carta et al., 2023; Zhou et al., 2024; Bai et al., 2024; Qi et al., 2025) or expert-trajectory finetuning (Zeng et al., 2024; Song et al., 2024). The step-level extensions that this paper compares against differ in the equivalence relation used to form the step group (Table 3). BiPACE (Wang et al., 2026a) induces the partition from the policy itself, clustering steps by cosine distance in the actor's hidden states, and its reproduction of GiGPO on ALFWorld with Qwen2.5-7B is the source of the singleton rates quoted in Section 1 (34.2% of step groups at iteration 10, 20.7% at iteration 140). Graph-based variants (Yuan et al., 2025a; Wang et al., 2026c; Cheng et al., 2026) share credit along a state-transition graph. Further step-level credit-assignment schemes for agents do not form state groups at all: stepwise progress attribution (Wang et al., 2025a), dense turn-level rewards (Wei et al., 2025), entropy-based criticality weighting (Shen et al., 2025), implicit step rewards (Liu et al., 2026), agent process reward models (Choudhury, 2025), grouped value proxies (Zhu et al., 2026), branching-score decision points (Wang et al., 2026b), hierarchical planner-executor credit (Peng et al., 2026) and segment-level advantages for reasoning (Guo et al., 2025); Zhang (2026) surveys the area. These are complementary to OADPO, which changes only the baseline. (A second, unrelated method also abbreviated ProGPO, progress-conditioned group policy optimization (Yang et al., 2026), reshapes advantages by the number of new states a rollout visits; we cite it for disambiguation.)

Weighted baselines and the distinction from ProGPO. All of the group-forming methods, including the weighted ones, are instances of $A(s,a) = R - \sum_k w_k b_k(s)$, a weighted state baseline; what distinguishes them is where the weights come from. Our contribution is not weighting per se but a basis in which the weights are *derivable* rather than tuned. The distinction from ProGPO deserves care, because inverse-variance weighting is the closest prior idea. Inverse variance combines several unbiased estimates *of the same quantity*, and its weight diverges as a group's size approaches one: the empirical variance goes to zero and the weight explodes, which is why the method carries a variance floor ϵ_{var} and a minimum group size. Shrinkage weights *different* quantities (orthogonal components of one function), and its weight degrades gracefully to zero as evidence vanishes, because the numerator $\hat{\sigma}_U^2$ vanishes with the evidence rather than the denominator. Shrinkage has entered critic-free RL before: EBPO (Han et al., 2026) pulls each prompt's group mean toward the batch-wide

mean with one scalar factor, the same ratio along the *prompt* axis; what is new here is shrinking in a basis with one block per part, which is tractable because orthogonality separates the MSE, together with the cluster correction that keeps the noise term valid for trajectory-structured data.

B PROOFS AND DERIVATIONS

Throughout the appendix, a *block* is one part f_U of the decomposition in Section 4.2, and *blockwise* means part by part.

B.1 FORMAL STATEMENTS DEFERRED FROM SECTION 4

Theorem 5 (Orthogonal decomposition; Hoeffding, 1948)**.** *Under Assumption 1, with*

$$f_U(x_U) = \sum_{W \subseteq U} (-1)^{|U|-|W|}\, \mathbb{E}[f \mid x_W] = \mathbb{E}[f \mid x_U] - \sum_{W \subsetneq U} f_W, \tag{5}$$

we have (i) $f = \sum_{U \subseteq [K]} f_U$ *with* $f_\emptyset = \mathbb{E}[f]$*;* (ii) $\mathbb{E}[f_U \mid x_W] = 0$ *whenever* $U \not\subseteq W$*;* (iii) $\langle f_U, f_W \rangle_{L^2(\nu)} = 0$ *for* $U \neq W$*;* (iv) $\mathrm{Var}(f) = \sum_{U \neq \emptyset} \sigma_U^2$ *with* $\sigma_U^2 = \|f_U\|^2$.

Assumption 2 (Block noise model)**.** The estimates $\hat{f}_U$ are unbiased for f_U; the errors $\varepsilon_U = \hat{f}_U - f_U$ satisfy $\mathbb{E}\|\varepsilon_U\|^2 = \eta_U$, are uncorrelated across blocks, and are independent of f.

Theorem 6 (Identifiability limit, formal version of Theorem 3)**.** *Let* $Z \in \mathbb{R}^{N \times p}$ *be the step-level design and* $y_i = h(t_i, T_{g(i)}) R_{g(i)}$ *for* any *deterministic* h*. With* $\tilde{Z}_h \in \mathbb{R}^{n_\tau \times p}$, $[\tilde{Z}_h]_{g,\cdot} = \sum_{i \in g} h_i z_i$*,*

$$Z^\top y = \sum_g R_g \Big(\sum_{i \in g} h_i z_i \Big) = \tilde{Z}_h^\top R, \tag{6}$$

so the identifiable subspace is $\mathrm{row}(\tilde{Z}_h)$*, of dimension at most* $\min(p, n_\tau)$*; any* $v \in \ker(\tilde{Z}_h)$ *has* $v^\top Z^\top y = 0$ *for every realization of* R*. Switching the target from the terminal reward (*$h \equiv 1$*) to the discounted return-to-go (*$h = \gamma^{T-t}$*) changes* $\tilde{Z}_h$ *and hence* which *directions are identifiable, but not the bound.*

Under Assumptions 1 and 2, Theorem 1 states that

$$\mathbb{E}\|b_w - f\|^2 = \sum_U \Big[(1 - w_U)^2 \sigma_U^2 + w_U^2 \eta_U \Big], \tag{7}$$

and the deployed plug-in estimator of Section 4.3 is

$$\hat{\sigma}_U^2 = \big(\|\hat{f}_U\|^2 / N - \eta_U \big)_+, \qquad \hat{w}_U = \frac{\hat{\sigma}_U^2}{\hat{\sigma}_U^2 + \eta_U}. \tag{8}$$

B.2 PROOF OF THEOREM 5

(i) By induction on $|U|$ using Equation (5): the Möbius inversion of $\mathbb{E}[f \mid x_U] = \sum_{W \subseteq U} f_W$ over the Boolean lattice with $\mu(W, U) = (-1)^{|U|-|W|}$. Taking $U = [K]$ and $\mathbb{E}[f \mid x] = f$ gives $f = \sum_U f_U$; $f_\emptyset = \mathbb{E}[f]$ is the $U = \emptyset$ case.

(ii) Let $U \not\subseteq W$ and pick $k \in U \setminus W$. Under Assumption 1, $\mathbb{E}[f_U \mid x_W] = \mathbb{E}_{x_{U \setminus W}}[f_U]$, and by construction f_U has zero mean in each of its arguments separately: $\mathbb{E}_{x_k}[f_U] = 0$ by the alternating-sum cancellation in Equation (5). Hence the conditional expectation vanishes.

(iii) For $U \neq W$, without loss of generality $U \not\subseteq W$. Then $\langle f_U, f_W \rangle = \mathbb{E}[f_U f_W] = \mathbb{E}\big[f_W\, \mathbb{E}[f_U \mid x_W] \big] = 0$ by (ii) and the tower property, since f_W is x_W-measurable.

(iv) Parseval, using (i) and (iii). □

B.3 Proof of Theorem 1

By Assumption 2, $\hat{f}_U = f_U + \varepsilon_U$ with $\mathbb{E}\varepsilon_U = 0$, $\mathbb{E}\|\varepsilon_U\|^2 = \eta_U$, and $\varepsilon_U \perp f$. Then $b_w - f = \sum_U \big[(w_U - 1)f_U + w_U\varepsilon_U\big]$. Expanding the squared norm, all cross terms across distinct U vanish: the $f_U f_W$ terms by Theorem 5(iii), the $\varepsilon_U\varepsilon_W$ terms by the uncorrelatedness in Assumption 2, and the mixed terms by $\varepsilon_U \perp f$ and $\mathbb{E}\varepsilon_U = 0$. This leaves Equation (7). Each summand $\phi(w_U) = (1 - w_U)^2\sigma_U^2 + w_U^2\eta_U$ is a strictly convex quadratic with $\phi'(w) = -2(1 - w)\sigma_U^2 + 2w\eta_U$, whose unique root is $w_U^\star$ as claimed; substituting back gives $\phi(w_U^\star) = \sigma_U^2\eta_U/(\sigma_U^2 + \eta_U)$. □

B.4 Proof of Corollary 2

$w = \mathbf{1}_{\{\emptyset\}}$ gives $\eta_\emptyset + \sum_{U\neq\emptyset}\sigma_U^2$ (only the constant block is kept, and its own estimation noise $\eta_\emptyset$ is paid; it cannot be dropped, since $\mathrm{MSE}^\star$ retains the term $\sigma_\emptyset^2\eta_\emptyset/(\sigma_\emptyset^2 + \eta_\emptyset)$), and $w \equiv 1$ gives $\sum_U \eta_U$. Both are feasible points of a strictly convex objective whose unique minimizer $w^\star$ lies in the open cube $(0, 1)^{2^{[K]}}$ whenever every block has $\sigma_U^2 > 0$ and $\eta_U > 0$; hence $\mathrm{MSE}^\star$ is strictly below the value at either vertex. If some blocks are degenerate ($\sigma_U^2 = 0$ or $\eta_U = 0$) the corresponding coordinate of $w^\star$ coincides with a vertex coordinate and strictness follows from the existence of at least one non-degenerate block, using the harmonic-mean identity $\sigma^2\eta/(\sigma^2 + \eta) < \min(\sigma^2, \eta)$. □

B.5 Proof of Theorem 3 and Corollary 4

Write $g(i)$ for the trajectory containing step i and $y_i = h_i R_{g(i)}$ with $h_i = h(t_i, T_{g(i)})$ deterministic given the trajectory. Then

$$Z^\top y = \sum_i z_i y_i = \sum_g R_g \sum_{i\in g} h_i z_i = \tilde{Z}_h^\top R.$$

The normal equations depend on y only through $Z^\top y$, so any v with $\tilde{Z}_h v = 0$ satisfies $v^\top Z^\top y = (\tilde{Z}_h v)^\top R = 0$ identically in R. Since $\tilde{Z}_h$ has n_τ rows, $\dim \mathrm{row}(\tilde{Z}_h) \le \min(p, n_\tau)$.

Corollary 4(i): the block U is unidentifiable iff every loaded trajectory-sum $\sum_{i\in g} h_i\tilde{q}_i$ vanishes, which is exactly the condition making the design-effect sum in Equation (9) zero. (ii) follows because $\|\hat{f}_U\|^2$ is then zero in expectation, so the positive part in Equation (8) truncates $\hat{\sigma}_U^2$ to 0 and $\hat{w}_U = 0$. (iii) Group centring by prompt removes one degree of freedom per prompt from the n_τ available trajectory-level observations. □

B.6 What breaks when Assumption 1 fails

Correlated features make the blocks non-orthogonal, so Theorem 5(iii) fails and the cross terms in the expansion of $\mathbb{E}\|b_w - f\|^2$ survive. Consequences, in decreasing order of severity: (a) the objective no longer separates, so $w^\star$ as given is a coordinatewise minimizer of a decoupled surrogate rather than the joint minimizer: it remains a valid shrinkage estimator but loses its optimality certificate; (b) the variance shares σ_U^2 no longer sum to $\mathrm{Var}(f)$, so they must not be reported as proportions; (c) the sequential (Type-I) orthogonalization we use makes the decomposition depend on the order in which blocks are residualized. What does *not* break: unbiasedness of the policy gradient, which requires only that b_w be a function of s_t (up to the in-sample fit of $\hat{b}$ on the current batch, whose per-record leverage is small; Appendix C).

B.7 The cluster-corrected noise term

Let Q_U be an orthonormal basis of the residualized design of part U. The noise term used in Equation (3) is

$$\eta_U = \frac{\mathrm{rank}(Z_U)\cdot \mathrm{DE}_U \cdot \sigma^2}{N}, \qquad \mathrm{DE}_U = \frac{1}{\mathrm{rank}(Z_U)}\sum_g \Big\| \sum_{i\in g} h_i\,\tilde{q}_i \Big\|^2, \qquad \tilde{Q} = (I - P_u)Q, \tag{9}$$

where g ranges over trajectories, DE_U is Kish's design effect (Kish, 1965), h_i is the target load ($h \equiv 1$ for the terminal reward, $h = \gamma^{T-t}$ for the return-to-go, normalized so that $\mathbb{E}[h^2] = 1$),

and P_u is the projection onto the prompt-group indicator, which matches the centering used by the advantage. The plug-in weight of Equation (8) is a positive-part James–Stein rule (Efron & Morris, 1973) fitted in-sample on the current batch; the truncation introduces a mild positive bias, which we report rather than correct. Before substitution, both $\|\hat{f}_U\|^2/N$ and η_U are smoothed across training steps by an exponential moving average with momentum 0.9 (Algorithm 1); w_U is recomputed from the smoothed values, not smoothed itself. Under an orthogonal design the rule is exactly blockwise ridge regression with penalty $\lambda_U = \mathrm{DE}_U \, d_U \sigma^2/\sigma_U^2$.

Why the correction must match the target. Computing DE_U under the terminal convention while regressing a discounted target changes it by an order of magnitude and in either direction: for one feature it gives 0.19 instead of the target-matched 9.31, below the 0.5 floor of D4, and for another 23.95 instead of 4.92.

B.8 Feature-selection checks

Every check below is a finite-sample safeguard for the same quantity, the plug-in weight $\hat{w}_U$. In the limit where Assumption 1 holds and all estimates are exact, every check is redundant, because shrinkage alone sends uninformative parts to $w \approx 0$. With finite samples, however, a purely soft rule selects features on its own estimation noise and keeps features that leak information about the outcome without describing the state. The seven checks, listed with their thresholds in Algorithm 3 and Table 4, implement three requirements.

- **Identifiable** (Corollary 4(i,iii)): features are binned to at most 6 levels; a part must have positive rank and $\mathrm{DE}_U \geq 0.5$ under the target-matched convention; and a budget on degrees of freedom caps greedy forward selection.
- **Approximately orthogonal** (Assumption 1): bins must be balanced within time strata, and a feature with Cramér's $V > 0.5$ against an already kept feature is discarded.
- **Signal rather than leakage**, the only family of checks that sees the target, evaluated on held-out folds: the held-out R^2 must be at least 0.005, a floor motivated by $w_U^\star > 1/2 \iff R_U^2 > \mathrm{rank}_U \, \mathrm{DE}_U/N$: with $\mathrm{rank}_U \, \mathrm{DE}_U$ of 2–12 for the deployed parts and batches of N step records in the low thousands, that bound is of order 0.005, and we use one common floor rather than a per-part one; and the feature must survive residualization against the step index *and* beat a baseline that uses the step index alone. Without this last check, "time until the episode ends" passes for state structure.

Selection is repeated under 8 prompt-level partitions ("salts"), and a feature enters the final *stable core* if more than half of them select it.

B.9 Cost

With $K \approx 5$ features, $c_k \leq 6$ bins and interaction order 2, the baseline has $O(10^2)$ parameters, and the per-step fit is one thin SVD per part on at most $N = 128 \times 50 = 6{,}400$ step records. There is no extra rollout, no second model and no extra forward pass. Algorithms 1 to 3 in Appendix C give the online loop, the fit and the selection stage. One guard is essential: if, after freezing, the design collapses to zero columns for three consecutive steps, which would silently degrade OADPO into GRPO, training aborts (Appendix C.5).

C IMPLEMENTATION DETAILS

C.1 THE ONLINE LOOP

Algorithm 1: OADPO advantage computation for one training batch

Input: rollout batch; frozen spec $\mathcal{S}$ (or $\varnothing$ during warm-up)
$\mathcal{R} \leftarrow$ build step records keyed on (trajectory, step) (sort by trajectory and step; attach parsed state; **never** a_t, reward, or future);
if $\mathcal{S} = \varnothing$ **then** cache $\mathcal{R}$; attempt freeze (Algorithm 3); **return** GRPO advantage, log `fallback`$= 1$;
$(\texttt{resid}, \mathrm{std}_g) \leftarrow$ group-center the RTG target by prompt; $X \leftarrow \mathcal{S}.\mathrm{transform}(\mathcal{R})$;
$h \leftarrow \gamma^{T-t}$;
fit $\leftarrow$ FITBASELINE$(X, \texttt{resid};\ \text{cluster} = \text{traj.},\ \text{load} = h,\ \text{center} = \text{prompt})$;
$(\|\hat{f}_U\|^2/N, \eta_U) \leftarrow \mathrm{EMA}_{0.9}(\text{fit})$; w_U via Equation (8); $\hat{b} \leftarrow \sum_U w_U \hat{f}_U$;
$A^{\text{step}} \leftarrow \mathbf{1}[\mathrm{std}_g > 10^{-8}](\texttt{resid} - \hat{b})/(\max(\mathrm{std}_g, 0.1\hat{\sigma}_{\text{ref}}) + 10^{-6})$; $A^{\text{ep}} \leftarrow$ GRPO advantage;
return $A^{\text{ep}} + w_{\text{step}} A^{\text{step}}$; `// `$w_{\text{step}} = 0 \Rightarrow$ `bit-identical to GRPO`

Two guards deserve emphasis: groups with $\mathrm{std}_g \le 10^{-8}$ receive $A = 0$ exactly as in GRPO, and the std floor is applied to the step-term denominator only, so the w_{step}=0 bit-exactness with GRPO survives it. This identity holds on the de-duplicated step batch that Algorithm 1 builds by keying records on (trajectory, step). The baseline $\hat{b}$ is fitted in-sample on the current batch; the mean leverage of a step record is $\sum_U \mathrm{rank}_U / N$ (at most $47/N$ on ALFWorld), so the optimism this introduces is small.

C.2 THE FIT

Algorithm 2: FITBASELINE: ANOVA decomposition with design-effect-corrected shrinkage

Input: binned features $X \in \{0, \ldots, c_k - 1\}^{N \times K}$; target R; cluster ids g (trajectory); target load h; centring ids u (prompt); interaction order 2
$\mathcal{K} \leftarrow \{\emptyset\} \cup \binom{[K]}{1} \cup \binom{[K]}{2}$;
Build the design Z: nominal factors get sum-to-zero effect coding ($c - 1$ columns, last level $= -1$); ordinal factors get Gram–Schmidt orthogonal polynomial contrasts; block U is the row-wise Khatri–Rao product of its factors;
$Q_{\text{acc}} \leftarrow [\,]$;
for $U \in \mathcal{K}$ ***in hierarchical order*** *(constant, then order 1, then order 2)* **do**
  $\tilde{Z} \leftarrow Z_U - Q_{\text{acc}}(Q_{\text{acc}}^\top Z_U)$; `// Type-I residualization`
  $(Q_U, \mathrm{rank}_U) \leftarrow$ thin SVD of $\tilde{Z}$, discarding singular values $\le$ `tol` $\cdot\, s_1\sqrt{N}$;
  $c_U \leftarrow Q_U^\top R$; $\hat{f}_U \leftarrow Q_U c_U$; append Q_U to Q_{acc};
end
$\hat{\sigma}^2 \leftarrow \|R - \sum_U \hat{f}_U\|^2 / \max(N - \sum_U \mathrm{rank}_U, 1)$;
Normalize the load so $\overline{h^2} = 1$; set $\tilde{Q}_U \leftarrow (I - P_u)Q_U$;
$\mathrm{DE}_U \leftarrow \mathrm{rank}_U^{-1} \sum_g \left\| \sum_{i \in g} h_i \tilde{q}_i \right\|^2$; $\eta_U \leftarrow \mathrm{rank}_U\ \mathrm{DE}_U\, \hat{\sigma}^2/N$; `// Equation (9)`
$\hat{\sigma}_U^2 \leftarrow (\|\hat{f}_U\|^2/N - \eta_U)_+$; $w_U \leftarrow \hat{\sigma}_U^2/(\hat{\sigma}_U^2 + \eta_U)$; `// Equation (8)`
return $\{\hat{f}_U,\ \mathrm{rank}_U,\ \mathrm{DE}_U,\ \eta_U,\ w_U\}, \hat{\sigma}^2$

Two details matter for reproduction. The constant block is never shrunk ($w_\emptyset = 1$), so OADPO retains GRPO's group mean exactly and only adds structure on top of it. Rank-deficient blocks return a zero coefficient vector rather than a division by zero.

C.3 SELECTION

Algorithm 3: SELECTFEATURES: seven gates, run once at freeze time

Input: warm-up records; salt s; $K_{\mathrm{cf}} = 5$ crossfit folds
Partition prompts by $\mathrm{md5}(u \,\|\, s) \bmod 3$ into $A/B/C$; assign crossfit folds by $\mathrm{md5}(u \,\|\, s) \bmod K_{\mathrm{cf}}$; $n_\tau^{\mathrm{eff}} \leftarrow n_\tau(K_{\mathrm{cf}} - 1)/K_{\mathrm{cf}}$;
Mine template features on split A observations only;
D1 bin on split A (equal-frequency, ≤ 6 levels, rare categories folded); drop constants;
D2 within each of 12 equal-frequency step strata, require bin balance (max share ≤ 0.70, or ≤ 0.95 for binary features, or a minimum count of 200);
D3 forward scan; drop f if stratum-weighted Cramér's $V(f, g) > 0.5$ for some kept g;
D4 compute $(\mathrm{rank}, \mathrm{DE})$ under the target-matched load; require $\mathrm{rank} > 0$ and $\mathrm{DE} \geq 0.5$;
$R^2_{\text{time}} \leftarrow$ out-of-fold R^2 of the step-bucket factor alone;
S1/S2 for each f: drop if $R^2_f < 0.005$, or if $R^2_{f,\text{detrended}} < 0.005$, or if $R^2_f \leq R^2_{\text{time}}$;
budget $\leftarrow \max(\lfloor n_\tau^{\mathrm{eff}}/\texttt{dof_ratio} \rfloor, 2)$;
S3 greedy forward selection on out-of-fold MSE among candidates whose ANOVA dof fits the budget; stop when the relative gain falls below 10^{-3} or no candidate fits;
return the specification, or $\varnothing$ (in which case training continues as GRPO)

C.4 HYPERPARAMETERS

Table 4: Complete configuration. The block weights w_U are computed, not configured; the values below govern the selection stage and the online guards. *ScienceWorld uses $\gamma = 0.8$ (Appendix E). †Freezing also needs 1500 cached trajectories, which at 128 rollouts per step takes at least 12 steps ($12 \times 128 = 1536$; 13 in the ScienceWorld rounds).

Parameter	Value	Parameter	Value
interaction order	2	warm-up steps before freeze	$\geq 10^{\dagger}$
max bins per feature	6	min cached trajectories	1500
binning	equal-frequency	min pos./neg. trajectories	30 / 30
design-effect floor $\mathrm{DE}_{\min}$	0.5	greedy tolerance	10^{-3}
redundancy threshold (Cramér V)	0.5	`dof_ratio` (online / offline)	20 / 8
max bin share (general / binary)	0.70 / 0.95	crossfit folds	5
EMA momentum ($\|\hat{f}_U\|^2/N, \eta_U$)	0.9	re-binning interval	40 steps
w_{step} in Equation (4)	1.0	degenerate-group threshold	10^{-8}
baseline target	RTG, $\gamma = 0.9^*$	step-term std floor	$0.1\times$ robust scale
PPO discount (untouched)	1.0	shrinkage rule	blockwise positive-part

C.5 THE FROZEN SPECIFICATIONS

ALFWorld (`alfworld_rtg09.json`). Features {`b4_carry`, `b4_loc`, `b3_distinct_frac`} with 2/6/6 bins; `b4_loc` maps 22 raw container types into 6 bins; $\mathrm{dof} = 1 + 11 + 35 = 47$ nominal, of which 46 are order > 0 (realized online rank of the order-> 0 blocks 33–46, Table 12, because sparsely populated bins are occasionally empty in a batch); observed warm-up marginals $[0.755, 0.245]$, $[0.255, 0.271, 0.069, 0.079, 0.156, 0.170]$ and 6 quantile bins respectively. The third feature is a step-index proxy (detrended $R^2 = -0.0004$, Appendix D): it entered the online selection of this run under an earlier gate ordering, before the mandatory detrending check of S2, and the current gates reject it. It adds $+0.0185$ to the offline R^2 ($\approx$0.130 at $\gamma = 0.9$, against 0.1118 ± 0.0084 for the two-feature specification the current gates return, Table 9); the two-feature specification has been trained only in the single-seed sensitivity arm of Appendix G.5.

WebShop (`webshop_match_rtg09.json`). {`m_opt_hit`} with 4 bins at cuts $[-1.0, 0.0, 0.5]$ and marginals $[0.000, 0.548, 0.195, 0.257]$; the first bin is unreachable because the -1 sentinel maps to bin 1, so after residualizing against the intercept the block has rank 2; this is why Table 12 reports `nominal_dof` $= 2$ rather than 3.

Engagement criterion. Every step logs `frozen`, `fallback`, the live design rank `nominal_dof`, its shrinkage-weighted counterpart `eff_dof`, the mean part weight per order and the baseline MSE reduction (Table 12). A reported OADPO run must satisfy `nominal_dof` $> 0 \wedge$ `fallback` $= 0 \wedge \bar{w} > 0$ at every step after warm-up.

D Feature catalogue and the no-leakage contract

D.1 Contract

Every feature is a function of the trajectory prefix (the observations up to and including the current one and the actions before it), the step index and the step budget, and of nothing else. Three boundaries hold by construction and were checked explicitly rather than assumed by convention.

1. **No reward, score, or success flag.** The record object physically carries the return (the fitting code needs it), and the extraction code may not read it. We verified that feature values are unchanged when the return field is permuted.
2. **No action a_t, and no future.** Features depend on $a_{<t}$ only. This is a correctness requirement, not a performance one: a feature depending on a_t makes b a function of the action and breaks the unbiasedness of the policy gradient. We also verified prefix causality by re-running each trajectory truncated at every t and checking that the prefixes are bit-identical.
3. **No privileged engine state.** On WebShop the terminal page prints the achieved score in plain text; even though such rows are already masked out of the record stream after termination, the parser detects the terminal page and carries the previous step's state forward without reading a character of it. The engine's goal dictionary (the literal input to the reward function) is off limits, as are ScienceWorld's score, subgoal index and win flag.

We allow one thing that deserves explicit disclosure: on WebShop, matching the *prompt text* of the instruction against the *prompt text* of the currently displayed item. Formally this is a sufficient statistic for $\mathbb{E}[R \mid s_t]$, and approximating $\mathbb{E}[R \mid s_t]$ is the definition of a baseline: being highly predictive is not leakage, *reading the outcome* is, and the two are separated mechanically by the truncated-replay test above. Readers who consider this too permissive should note that removing the M family removes WebShop's only surviving feature, and WebShop is the environment on which the four arms end closest together with Llama-3.2-3B and SmolLM3-3B.

D.2 Catalogue

Table 5 lists the candidate feature families and marks the ones that entered a deployed specification.

D.3 Selection outcomes

On ALFWorld, `b4_carry` and `b4_loc` are selected by 8/8 salts (single-feature out-of-fold R^2 of 0.0765 and 0.0437 on the group-centred residual at $\gamma = 0.9$; design effects 4.92 and 2.36); `b4_subgoal` dies at D3 in all 8 salts (Cramér's V (Cramér, 1946) $\in [0.80, 1.00]$ against `b4_carry`: it is a relabelling of an already-kept feature); `b3_phase` dies at S2 with a detrended $R^2 \approx 0$; and `b3_distinct_frac` dies at S2 with a detrended R^2 of -0.0004, that is, it encodes the step index and nothing else.

On WebShop the entire trajectory-shape family (18 candidates) dies at D4 with design effects of 0.17–0.46, below the 0.5 floor: those features carry almost no loaded within-prompt contrast, so their parts are too weakly identified to use at this sample size. The one survivor, `m_opt_hit` (7/8 salts, single-feature $R^2 = 0.0682$, DE $= 0.897$; as the selected one-feature specification, 0.0678 ± 0.0100 over salts, Table 9), passes under both target conventions (terminal and return-to-go) and is the highest-ranked feature under each. On ScienceWorld no single feature passes S1: the engine-derived progress features score 0.0004 and 0.0030, and `b4_loc` and `b3_since_new` score 0.0019 and 0.0028 individually, yet are selected jointly in 8/8 salts with a held-out R^2 of 0.1250 ± 0.0097 on the detrended residual (Section 5.4). Single-feature R^2 values are not comparable across selection rounds, because the candidate set determines the residual each feature is scored against.

Table 5: Candidate feature families. All are prefix-causal functions of the observation and action history. Features marked **selected** pass the current gates and entered a deployed specification; `b3_distinct_frac` entered the deployed ALFWorld specification under an earlier gate ordering (Appendix C.5).

Family	Feature	Environments	Definition (sketch)
B1	`b1_change_mag`	all	composite of observation-change indicator, token Jaccard distance, and clipped new/lost token counts; excludes the failure-keyword flag
B2	`b2_t`j	all	slot content of the j-th template mined from split-A observations; `ABSENT` when no sentence matches
B3	`b3_phase`	all	$\min(3, \lfloor 4t/\text{budget} \rfloor)$
	`b3_since_new`	all; ScienceWorld (joint route, Section 5.4)	steps since the last novel observation
	`b3_distinct_frac`	all; ALFWorld (deployed, earlier gates)	distinct observations so far $/(t+1)$
	`b3_repeat5`	all	repeats of a_{t-1} among the last five actions
	`b3_prev_verb`	all	first token of a_{t-1}
B4	**`b4_carry`**	ALFWorld **(selected)**	whether the agent is holding an object, from a prefix state machine over *attempted* take/put actions (the action text, not the environment's confirmation; identical for all arms)
	`b4_loc`	ALFWorld **(selected)**; ScienceWorld (joint route, Section 5.4)	container *type* of the last arrive/open event; instance numbers discarded, 22 types folded to 6 bins
	`b4_subgoal`	ALFWorld	4-level composite of carry state and placement count
M	**`m_opt_hit`**	WebShop **(selected)**	clicked options matching a required option, over options clicked; -1 before any click
	`m_title_cov`	WebShop	instruction-description tokens covered by the current item title
g	`g_prefix`	ScienceWorld	monotone ordered-subgoal progress, read from text
	`g_opt`	ScienceWorld	fraction of six textually observable optional goal predicates satisfied; deliberately non-cumulative so it cannot proxy for t

E EXPERIMENTAL PROTOCOL

Seeds. Every arm of the main comparison on ALFWorld, WebShop and ScienceWorld and of the attribution ablation is trained with three seeds (0, 1, 2), and each such number is the mean of the three seeds' values at the stated evaluation point; only the sensitivity arms of Appendix G.5 use seed 0 and are marked as such.

E.1 EVALUATION PROTOCOLS

Split sizes: ALFWorld train 3553 / `valid_seen` 140; WebShop train goals 500–6909 / eval goals 0–499.

ALFWorld. `valid_seen`, randomly sampled at a validation batch size of 128 from the 140 available games, every 10 steps (31 evaluation points per arm). GiGPO's published ALFWorld numbers use the same `valid_seen` default (its code does not override the evaluation dataset key), so the protocol is shared within the RL line. The Llama-3.1-8B and SmolLM3-3B ALFWorld rounds instead evaluate all 140 `valid_seen` games every 10 steps.

WebShop. Goals 0–499, a deterministic full sweep in 4 batches of 125, every 20 steps. Wording we use throughout: *following verl-agent (the official GiGPO implementation), we evaluate on the official test split (goals 0–499) and train on goals 500+, the original evaluation range having been folded into the training pool by upstream default; all four algorithms share one environment wrapper and one evaluation protocol.*

ScienceWorld RL round. Every evaluation sweeps the full dev set (`dev-1740`: all 1740 dev variations of the 30 tasks, in a fixed order), every 10 steps for 250 steps, with generation temperature 0.4; each arm is trained with three seeds, and the OADPO arm uses the joint specification of Appendix F.3 with $\gamma = 0.8$. All arms start from the BC step-198 checkpoint, whose `dev-1740` value $B = 27.5\%$ fixes the pre-registered success line $B{+}15$ points $= 42.5\%$; the OADPO arm spends its first 13 steps in warm-up (`fallback`$= 1$, *i.e.* GRPO) while the fitting window fills, and curves are read from step 14 on.

Llama-3.1-8B rounds. All 8B arms train for 250 steps and evaluate every 10 steps (26 points). The ALFWorld-8B OADPO arm re-uses the deployed three-feature specification. WebShop-8B uses the deterministic test-split sweep above but starts *from the base model* rather than a behaviour-cloning checkpoint (step-0 success is 1.2%), so its column is not comparable to the 3B WebShop column in either level or per-step cost. ScienceWorld-8B follows the `dev-1740` protocol above from the 8B BC step-198 checkpoint (32.1%).

SmolLM3-3B rounds. The third backbone is SmolLM3-3B (Bakouch et al., 2025), a different model family from the Llama pair, run with its reasoning mode switched off (`/no_think`) so that responses are actions rather than chains of thought. All four arms use the 3B main-round training configuration ($G = 8$, $16{\times}8$ episodes per step, 2 GPUs) and evaluate every 10 steps. ALFWorld-SmolLM3 trains for 300 steps from the bare instruct model and evaluates on the deterministic sweep of all 140 `valid_seen` games; its OADPO arm re-uses the deployed three-feature specification, re-binned online as in the 3B round. WebShop-SmolLM3 trains for 300 steps from a SmolLM3 behaviour-cloning checkpoint taken at the *same* step (155) and on the same data as the Llama-3.2-3B BC start, without re-selecting it (step-0 test success 60.0%), and evaluates on the deterministic 500-goal test sweep. ScienceWorld-SmolLM3 trains for 250 steps from the position-matched SmolLM3 BC step-198 checkpoint (step 0: 24.6–25.8% across arms) under the `dev-1740` protocol, with the same $\gamma = 0.8$ joint specification as the 3B round.

F SCIENCEWORLD: THE FULL RECORD

F.1 EVERY AXIS WE VARIED

Table 6: Every data-generation and modelling axis varied, and what it changed. All values are out-of-fold R^2 against the group-centred residual. The pipeline produces no specification under the official gates in any configuration, so no arm was trained under this marginal screen.

Axis varied	Setting	Outcome	Reading
Baseline discount γ	five values, 0.80–0.99 (effective horizons $1/(1-\gamma)$ of 5–100 steps)	$\leq 1/8$ salts select	not a horizon problem
Feature family	engine-derived progress coords	pass D1–D4, die at S1	not a coordinate problem
Group size G	8 vs 16	0.0067 vs 0.0062	no effect, direction reversed
Behaviour policy strength	stronger vs weaker BC	0.0099 vs 0.0067	*more* contrast, *less* signal
Degrees-of-freedom budget	`dof_ratio` 8 vs 4	no specification either way	budget is not binding
Gates removed entirely	skip S1/S2	0.0596 ± 0.0077, 8/8 salts	the gate is binding
Gates removed + task-conditioned	skip S1/S2	0.0866 ± 0.0172	gate still binding

Three of these deserve comment. The **engine-derived feature family** was read off the environment's own scoring function and pre-registered before submission; both features passed every design gate and died at the signal gate. One of them, a count of satisfied optional subgoals, varies within group 54.1% of the time (higher than WebShop's selected feature at 34.4%) and still carries no signal, because the engine's success criterion is a definitional identity on the *ordered* progress index alone, so the binary outcome contains no information about optional subgoals at all. "Varies but is irrelevant" is a distinct failure from "relevant but invariant", and this family gave us one clean instance of each.

The **behaviour-policy cell** corrected a standing assumption of our own. A weaker behaviour-cloning checkpoint produces more within-group success/failure contrast (63.1% vs 56.6%) and therefore a larger *information budget* in the sense of Corollary 4, yet the explained variance went *down* (0.0067 vs 0.0099). Within-group contrast is a necessary condition for extractable signal, not a sufficient one. (This comparison is partly confounded by trajectory length, 84.2 versus 75.6 steps, which reduces discount coverage by 11%, so we report it as a decomposition still pending rather than a clean result.)

The **gates-removed** cells localize the failure. With S1/S2 skipped the pipeline produces specifications in 8/8 salts, so the data and the estimator are not the obstruction; and once the gates are gone the degrees-of-freedom budget binds immediately (7 of 8 salts terminate exhausted), which is Corollary 4(iii) one layer down. The three constraints are serial: gate, budget, data volume, and under the official gates only the first binds.

F.2 WHERE THE SIGNAL LIVES

Table 7 was run against a decision rule fixed before the job: joint-layer share $J \geq 60\%$ implies the screening architecture is at fault. Measured $J = 82\%$ ($\gamma = 0.8$) and 70% ($\gamma = 0.9$), both clearing it, with main effects contributing $+0.0054$ and $+0.0053$ (at the 0.005 floor of S1). Two independent cross-checks: the base model R_0 (0.0709) agrees to 0.0001 with the same quantity computed by a different probe under a different estimator (ridge versus shrinkage), which closes the "the base model is an implementation artifact" escape route; and the nested selection-honest evaluation reports optimism of $+0.008$ and exactly 0.000 at the budgeted $k = 4$ under two task groupings, because the greedy path is identical across all outer folds for $k \leq 3$.

Table 7: Where the signal lives. R_0 contains only task identity, time and their interaction; each row adds a layer. **Main effects contribute $+0.005$ (at the S1 floor), while the interaction layers carry 82% (70%) of the gain over R_0**, both clearing the pre-registered 60% decision line. R_0 was independently reproduced by a second probe with a different estimator, agreeing to 0.0001.

Model	$\gamma = 0.8$		$\gamma = 0.9$	
	R^2	Δ	R^2	Δ
R_0: task, time, task×time	0.0709	ref.	0.0479	ref.
R_1: + feature main effects	0.0763	$+0.0054$	0.0532	$+0.0053$
R_4: + all interaction layers	0.1016	$+0.0253$	0.0654	$+0.0122$
Joint-layer share J	**82%**		**70%**	

F.3 The joint route and the four-arm RL round

Selection. A later, pre-registered route replaces the marginal screen with joint selection: interaction blocks are scored together on the detrended residual, under the same design gates and the same dof budget. It produces a specification in 8/8 salts with held-out $R^2 = 0.1250 \pm 0.0097$; the previously dominant spurious family drops from 4/8 selections to 0/8. This R^2 is measured on the detrended residual, whereas Tables 6 and 7 use the group-centred residual, so the two sets of values are not directly comparable.

The four-arm round. All four algorithms trained for 250 steps from the BC step-198 checkpoint under the protocol of Appendix E. OADPO's arm is a genuine OADPO arm (after the 13-step warm-up, `fallback`$= 0$ for all remaining steps, and task out-of-vocabulary occupancy is 0 throughout).

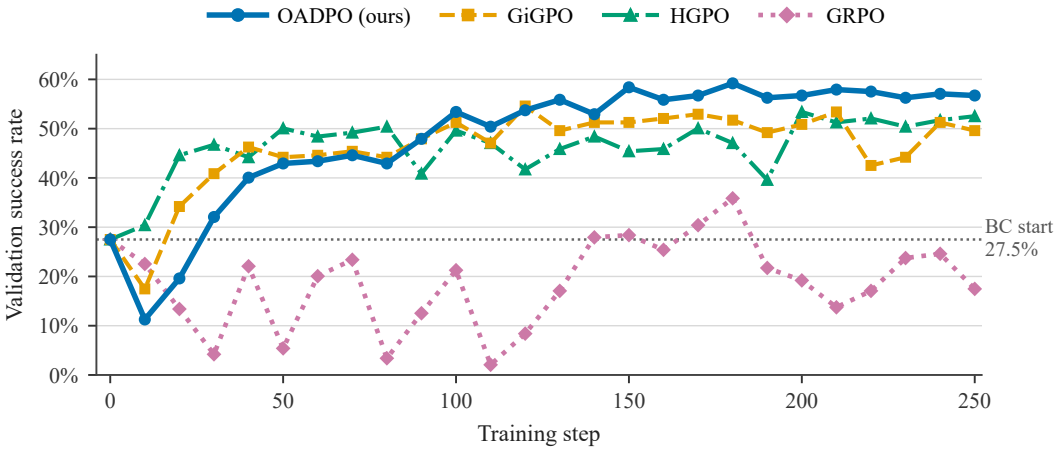

Figure 4: ScienceWorld validation success rate (full dev set, 1740 variations), one curve per algorithm (mean over three seeds); points every 10 steps, no smoothing. The OADPO arm's first 13 steps are GRPO warm-up.

Table 8: ScienceWorld RL round (3B): mean over three seeds per algorithm, every evaluation on the full dev set (dev-1740). BC start $B = 27.5\%$; success line $B+15$ points.

	GRPO	GiGPO	HGPO	OADPO
Final point, step 250	17.5%	49.6%	52.5%	**56.7%**

Outcome. OADPO ends highest of the four algorithms at the final point (56.7%, 4.2 points above HGPO); it is also the only arm that stays at or above 55% late in training (12 of its last 13 evaluation points, against none for GiGPO and HGPO).

The Llama-3.1-8B and SmolLM3-3B rounds. The same four arms and protocol were run from the 8B and the SmolLM3 BC step-198 checkpoints (Appendix E), carrying the joint specification selected on Llama-3.2-3B data over unchanged. Final points are in Table 11: at 8B OADPO ends highest (47.9%, 0.8 points above HGPO), and with SmolLM3-3B HGPO ends 0.5 points above OADPO (50.4% against 49.9%), the one ScienceWorld round in which OADPO does not end highest. Both OADPO arms engage after warm-up (`fallback`$= 0$, `frozen`$= 1$).

Fairness accounting. Two checks pass: the fitted baseline depends on the state features alone (no action, reward or future information; Appendix D), and there is no leakage (ScienceWorld is the one environment of our three with measured-disjoint train/dev/test splits, and dev pools never enter fitting or selection).

G Extended results

G.1 Offline selection across the discount grid

Table 9: Out-of-fold R^2 of the specification that the selection stage returns when run before RL on logged rollouts, against the group-centred residual, for the two environments deployed with the marginal screen, by baseline-target discount γ, over 8 independent prompt-level partitions ("salts"); $k/8$ = number of salts producing a specification. This is a screening check only: the deployed specifications come from the online selection at the end of warm-up (Algorithm 3) in the Llama-3.2-3B runs and are reused for the other backbones (Appendix E). At $\gamma = 0.9$ the offline selection returns the deployed specification on WebShop; on ALFWorld it returns the two-feature specification, while the online selection of the deployed run, made under an earlier gate ordering, also kept a step-index proxy (Appendix C.5). ScienceWorld appears nowhere in this table by design: its specification comes from the joint-selection route of Appendix F, not the marginal screen documented here.

γ	0.80	**0.90**	0.95	0.97	0.99
ALFWorld	8/8 0.1395	**8/8** $\mathbf{0.1118 \pm 0.0084}$	8/8 0.0756	8/8 0.0470	2/8 0.0115
WebShop	7/8 0.1466	**7/8** $\mathbf{0.0678 \pm 0.0100}$	7/8 0.0363	7/8 0.0284	7/8 0.0228

G.2 Endpoints and compute for the main round

Table 10 adds to the final evaluation point of Table 1 the first crossing of 90% and the measured training compute.

Table 10: The main round (300 steps, 3B) in full. Final is the last evaluation point, a mean over three seeds; compute sums per-step timings over 300 steps. Best in bold.

		Success rate (↑)		Training compute (↓)		
Environment	Method	Final	First $\geq 90\%$	Hours	s/step	vs. GRPO
ALFWorld	GRPO	59.6%	never	30.0	360	ref.
	GiGPO	91.7%	190	**27.3**	**327**	$\mathbf{-9\%}$
	HGPO	90.9%	200	28.0	337	-6%
	OADPO (ours)	**96.4%**	**170**	28.0	336	-7%
WebShop	GRPO	78.9%	never	**5.4**	65	ref.
	GiGPO	**81.7%**	never	6.2	75	$+16\%$
	HGPO	79.9%	never	7.3	88	$+36\%$
	OADPO (ours)	80.9%	never	**5.4**	**64**	$\mathbf{-0.2\%}$

G.3 The other two backbones

At 8B, OADPO ends 0.7 points above GiGPO on ALFWorld and, unlike the 3B round, highest of the four on WebShop (81.8% against 81.2%); both OADPO arms pass the engagement checklist at every step after warm-up (`fallback`$= 0$, `frozen`$= 1$; ALFWorld: median `nominal_dof`

Table 11: Final success rate (%) and training compute (hours, summing per-step timings) for the Llama-3.1-8B and SmolLM3-3B rounds (Appendix E). 8B rounds: 250 steps; SmolLM3: 300 steps on ALFWorld and WebShop, 250 on ScienceWorld; evaluations every 10 steps, final = last evaluation point. All finals are means over three seeds. Best in bold within a column.

	ALFWorld		WebShop		ScienceWorld	
Method	Final	Hours	Final	Hours	Final	Hours
Llama-3.1-8B-Instruct						
GRPO	92.9	–	76.6	23.9	24.2	–
GiGPO	96.4	–	81.2	15.6	37.5	–
HGPO	95.0	–	77.8	**15.3**	47.1	–
OADPO (ours)	**97.1**	–	**81.8**	16.3	**47.9**	–
SmolLM3-3B						
GRPO	84.3	26.0	76.6	**4.3**	37.5	37.3
GiGPO	93.6	22.7	78.0	4.5	46.3	34.0
HGPO	93.6	26.7	79.0	5.4	**50.4**	**33.3**
OADPO (ours)	**95.0**	**21.0**	**79.8**	4.6	49.9	38.6

≈46, median baseline-MSE reduction 26%; WebShop: median $\bar{w}$ over order-1 blocks 0.99, median baseline-MSE reduction 17%). With SmolLM3-3B, a different model family, the final-point ordering recurs: OADPO ends highest on ALFWorld (95.0% against 93.6% for GiGPO and HGPO) and on WebShop (79.8% against 79.0% for HGPO), margins of 1.4 and 0.8 points, while on ScienceWorld HGPO ends 0.5 points above OADPO (Appendix F.3). The SmolLM3 WebShop round reproduces the length behaviour of the 3B round (Figure 5): HGPO's mean response grows from 39 to 188 tokens and GiGPO's from 42 to 79, while GRPO and OADPO stay at 42–48; the ALFWorld arms show no such split (77–100 tokens late in training). The SmolLM3 OADPO arms engage at every step after warm-up on all three environments (median $\bar{w}$ over order-1 blocks 0.87 on ALFWorld and 1.00 on WebShop, median baseline-MSE reduction 20% and 41%).

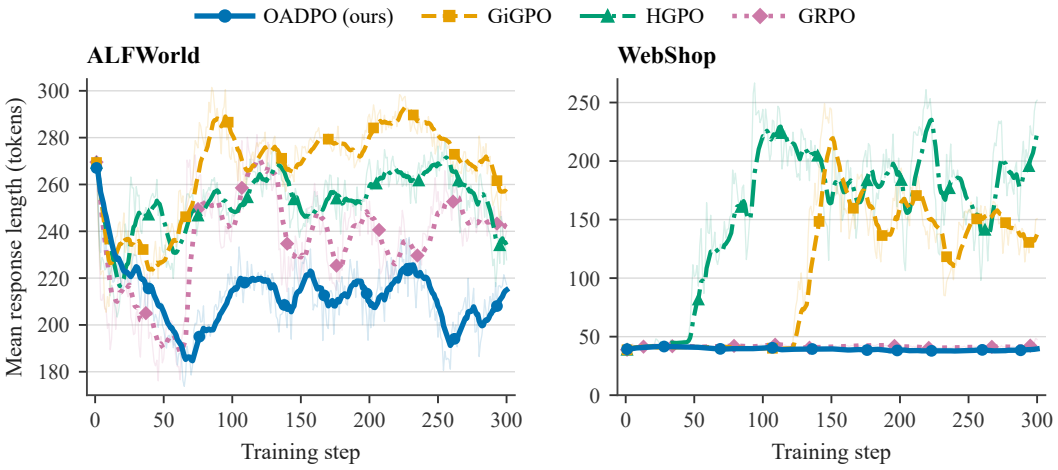

Figure 5: Mean generated response length against training step. Faint lines are the raw per-step series; heavy lines are an exponential moving average (span 12). On WebShop, HGPO undergoes a step-change increase at around step 45 and GiGPO at around step 120, after which both oscillate at a high level, while GRPO and OADPO stay flat at the ≈ 40 tokens a well-formed `search[·]` or `click[·]` action requires. On ALFWorld all four arms end shorter than they start, and OADPO ends shortest (≈215 tokens against 234–257 for the other arms, moving average at step 300).

On WebShop, the compute differences of Table 10 follow these length curves rather than the cost of the advantage estimators: OADPO and GRPO run at 64.5 and 64.6 seconds per step.

G.4 ENGAGEMENT DIAGNOSTICS IN FULL

Every OADPO run reported in this paper carries the checklist of Table 12.

Table 12: Engagement diagnostics for the two OADPO arms over all steps after warm-up. `nominal_dof` $= \sum_{|U|>0} \mathrm{rank}_U$ is the number of live design columns and `eff_dof` $= \sum_{|U|>0} w_U \, \mathrm{rank}_U$ its shrinkage-weighted counterpart; the gap between them is the shrinkage doing its job. `fallback`$= 1$ would mean the run reverted to GRPO; parentheses give bins per feature.

Quantity	ALFWorld OADPO	WebShop OADPO
Frozen specification	`b4_carry`/`b4_loc`/`b3_distinct_frac` (2/6/6)	`m_opt_hit` (4)
`nominal_dof` (median; range)	46 (33–46)	2 (constant)
`eff_dof` (median)	27.27	2.00
$\bar{w}$ over order-1 blocks (median)	0.875	0.998
Baseline MSE reduction (median)	0.170	0.369
`fallback` / `frozen`	0 / 1 (after warm-up)	0 / 1 (after warm-up)

Both arms engaged at every step after warm-up. The MSE-reduction row directly measures the claimed mechanism: relative to GRPO's group-mean baseline, the fitted baseline cuts mean-squared prediction error by a median of 17% on ALFWorld and 37% on WebShop.

G.5 ABLATION ON ALFWORLD: PROTOCOL, PER-SEED VALUES AND DIAGNOSTICS

Design. Llama-3.2-3B on ALFWorld under the main-round training configuration ($G = 8$, 16×8 episodes per step, 2 GPUs, learning rate 10^{-6}, KL 0.01, at most 50 environment steps per episode), for 200 optimization steps; `valid_seen` (140 games, temperature 0.4) is evaluated every 10 steps and the final point is step 200. Because the run is shorter than the main round, levels are not comparable with Table 1. *Attribution arms* (seeds 0, 1, 2): the full method ($\gamma = 0.9$, $w_{\mathrm{step}} = 1$, deployed three-feature specification, re-binned every 40 steps); the same with the fitted baseline set to zero before it enters the advantage ($\hat{b} \equiv 0$: the baseline is still fitted every step, so its quality can be read from the "shadow" diagnostics, but the step term is the standardized discounted target alone); $w_{\mathrm{step}} = 0$, which is GRPO bit-for-bit on the de-duplicated batch (Appendix C). *Sensitivity arms* (seed 0 only): one design choice changed at a time from the full method. Paired differences are taken within a seed before averaging. The pre-registered rules, fixed before the arms ran, were: a three-seed paired difference counts as established only if its mean over seeds exceeds 4.5 points, and is called unresolved if its magnitude is below 3 points; single-seed differences below 5 points are not interpreted. The seed-0 full arm is the first 200 steps of an earlier 250-step run. All arms pass the engagement checklist (`frozen`$= 1$, `fallback`$= 0$ after warm-up; `zero_baseline` $= 1$ only in the $\hat{b} \equiv 0$ arm; step-term standard deviation identically 0 only in the $w_{\mathrm{step}} = 0$ arm).

Table 13: Attribution arms per seed (final point, step 200, %), the mean over seeds, and paired differences.

Arm	Seed 0	Seed 1	Seed 2	Mean
OADPO (full)	90.7	89.3	90.0	90.0
$\hat{b} \equiv 0$ (discounted step target only)	86.4	82.1	83.6	84.0
$w_{\mathrm{step}} = 0$	67.9	48.6	72.9	63.1
OADPO $-$ ($w_{\mathrm{step}} = 0$)	$+22.8$	$+40.7$	$+17.1$	$+26.9$
OADPO $-$ ($\hat{b} \equiv 0$)	$+4.3$	$+7.2$	$+6.4$	$+6.0$
($\hat{b} \equiv 0$) $-$ ($w_{\mathrm{step}} = 0$)	$+18.5$	$+33.5$	$+10.7$	$+20.9$

Reading the attribution arms. The step term carries most of the gain: removing it costs 26.9 points on average and lowers success on every seed, while the $\hat{b} \equiv 0$ and full arms differ by 6.0 points, with every seed favouring the fitted baseline. The two curves cross between steps 60 and 70 and stay separated thereafter (Figure 3b). The baseline is fitted equally well in both arms (variance

explained 0.27 against 0.29 shadow; effective degrees of freedom 31 against 29), so the positive effect is not an artifact of fit quality. The claim that part of the ALFWorld gain comes from the fitted control variate (the orthogonal decomposition) is therefore supported.

Where the signal lives. The variance explained by the fit and its effective degrees of freedom rise over training in every arm that trains well ($0.02 \to 0.27$ and $15 \to 31$ for the full arm) and stay low in the $w_{\text{step}} = 0$ arm (0.10, 16), whose policy ends near 60% success with episodes of about 31 steps: most discounted targets of such a policy are near zero or negative, and state features cannot explain them. The data therefore contain a fittable state signal only once the policy is strong, which is why the attribution must be paired within a seed rather than read from single runs. The standard-deviation floor of the step term (Algorithm 1) is hit by 30–44% of records in the first 40 steps of every two-term arm and by 2–16% at the end, which is why early separations between arms are not interpreted.

Reading the sensitivity arms. Table 2 (bottom) reports the seed-0 finals; Section 5.2 discusses them. Three mechanistic observations support the reading there. The advantage scale grows roughly linearly with w_{step} ($\sigma_A/\sigma_{A^{\text{ep}}}$ of $1.00/1.43/2.29/3.69$ for $w_{\text{step}} = 0/0.5/1/2$) while $w_{\text{step}} = 2$ reaches every success threshold at the same step as $w_{\text{step}} = 1$, so the deployed value sits on a plateau. The single-term advantage puts 70% of its records in the first 40 steps on the standard-deviation floor (in an all-failure group the discounted targets differ only by the 0.1 invalid-action penalty of Equation (1), so the within-group standard deviation collapses), against 32% with the rollout term, which is why it reaches each threshold 60–80 steps later. Third, the arm that drops the second-order parts ends below the arm that never uses the fit (81.4% against 86.4%), so a partial baseline can cost more than none: within a prompt the residual target is dominated by its time curve γ^{T-t}, and the order-1 subspace of the deployed specification is dominated by time-correlated structure (one of the three features is a step-index proxy, Appendix C.5, and the main effects of the other two are largely progress-correlated), so a main-effects-only fit subtracts the target's time trend from the step term while contributing none of the state-conditional parts, removing the credit gradient the discounted target exists to provide; the interaction parts restore it. This is the same structure the S2 detrending check exists to exclude, and the deployed order-2 specification keeps the state-conditional part of the fit. Like every row of this block, the cell is single-seed, so it is offered as a mechanistic reading, not an established effect.

G.6 Extended limitations

Section 6 names the two limitations that matter most; this section states them, and the others we know of, in full.

The task must decompose into features someone can name. OADPO credits a step only through features that describe its state, and the candidate features are written by hand for each environment (Appendix D). Writing them requires knowing in advance which facts about the state decide the outcome: whether the agent holds the target object and where it is in ALFWorld, or whether the clicked options match the required ones in WebShop. When this decomposition is unknown, or cannot be read from the observation text, no part beyond the overall mean carries signal, the selection stage keeps no feature, and OADPO reduces to GRPO. The selection among candidates is automatic; the candidate pool is not, and learning features from the observation text remains future work.

Only a few parts can be estimated. With an outcome-only reward, the data identify at most as many directions as there are trajectories, however many steps they contain (Theorem 3); on ALFWorld, $19{,}473$ step records logged before RL give only $N_{\text{eff}} = 308$ (Section 4.5), and an online batch of 128 rollouts over 16 prompts gives 112. The baseline is therefore limited to a few features with at most six bins each and interactions up to order two, and state structure that needs more features, finer bins or higher-order interactions cannot be estimated at the batch sizes we use.

Signal that lives only in interactions is hard to select. A candidate feature is kept only if it is predictive on its own. When the signal lies in feature interactions rather than main effects, as on ScienceWorld, the one-feature-at-a-time screen keeps nothing, and a separate joint-selection route is needed (Appendix F).

The closed-form weights are optimal only for independent features. The optimality of $w^\star$ (Theorem 1) and its advantage over both corners (Corollary 2) assume independent features (Assumption 1). Real features are correlated: the redundancy check keeps them close to independent, but the deployed weights are only approximately optimal, by an amount we do not quantify, and the sequential orthogonalization makes the decomposition depend on the order of the blocks (Appendix B.6). The unbiasedness of the policy gradient does not depend on this assumption.

The features are fixed after warm-up. The specification is selected once and then frozen, and the other two backbones reuse the Llama-3.2-3B specifications unchanged (Appendix E); only the bin edges are refreshed. The fittable state signal grows as the policy improves (Appendix G.5), and the states that decide the outcome can shift with it, but a frozen specification cannot add a feature that becomes informative later.

Part of the gain comes from changing the objective. The step target $\gamma^{T-t}R$ rewards finishing sooner as well as finishing. On ALFWorld, where shorter rollouts succeed more often ($\mathrm{corr}(T, R) = -0.686$), the discounted target used without the fitted control variate accounts for 20.9 of the step term's 26.9 points (Appendix G.5); on tasks where rollout length and success are unrelated, this part of the gain need not carry over.

Scope. We evaluate three text environments, three open-weight backbones of 3B to 8B parameters with three seeds each, and one hand-specified feature pool per environment. Larger models and environments whose observations are not text are untested.