PAIRWISE WORKFLOW POLICY OPTIMIZATION FOR TOOL-USING LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using language agents can automate complex tasks, but reinforcement learning treats different execution orders as distinct experiences even when they produce identical outcomes, introducing noise unrelated to task quality. We introduce Pairwise Workflow Policy Optimization (PWPO), which verifies that swapping two adjacent tool calls preserves their outputs and resulting state, then optimizes the summed probabilities of both complete execution histories. In deterministic environments with restorable states, this aggregation preserves the expected on-policy gradient and removes variation within verified pairs; our analysis specifies when this reduction offsets additional computation. Across AppWorld-Atomic, ALFWorld, and WebShop under matched rollout budgets, PWPO achieves the highest means among evaluated methods on all 16 main metrics, improving AppWorld-Atomic task completion over G2PO by 2.32–2.64 percentage points. These results support learning from verified workflow alternatives while retaining the agent's sequential deployment interface.

1 INTRODUCTION

Tool-using language agents can turn natural-language goals into actions across software services (Trivedi et al., 2024) and interactive websites (Yao et al., 2022). Their practical value depends on reliably completing multistep tasks while learning from limited interaction budgets. Reinforcement learning offers a direct way to improve task success, but delayed feedback and diverse execution histories make it difficult to extract stable learning signals from each rollout.

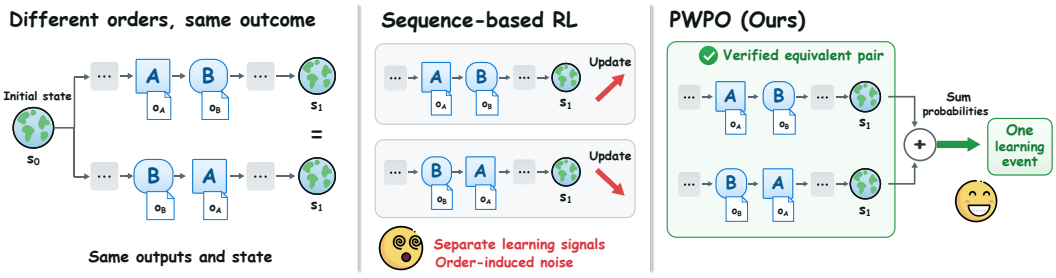


Figure 1: **Different execution orders, one learning event.** Two independent tool calls can return the same outputs and reach the same state in either order, yet induce different policy updates. PWPO verifies an adjacent swap and sums the probabilities of the two complete trajectories, treating them as one learning event.

Recent progress has focused on credit assignment: determining which decisions deserve reinforcement when rewards arrive only after many interactions. Group-relative objectives compare sampled outcomes to estimate advantages (Shao et al., 2024), while step-level methods use repeated states to evaluate local decisions (Feng et al., 2025). Context-aware grouping accounts for different histories leading to the same observation (He et al., 2026), and graph-based estimation shares evidence across trajectories to assess task progress (Wang et al., 2026a). These advances improve reward attribution. A separate source of variation comes from the ordering of otherwise equivalent work.

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.

Two independent queries, for example, may retrieve the same fixed records in either order while preserving the environment and reward. The model conditions on the preceding interaction history, so these executions can receive different probabilities and induce different gradient directions. Even accurate reward attribution can therefore leave update variation unrelated to task quality. Treating histories as interchangeable requires matching observations and preserving consequential dependencies as well as terminal reward. Inferring partial orders exposes flexibility in successful workflows (Li et al., 2026). We ask how policy optimization can use this flexibility to reduce order-induced variation while preserving the reward objective, and when the reduction offsets the added training cost.

We introduce *Pairwise Workflow Policy Optimization* (PWPO), which treats a trajectory and a verified alternative ordering as one learning event and optimizes their combined probability (Figure 1). Considering one local alternative at a time makes this idea tractable in deterministic tool environments with restorable state, while retaining the agent's original interface at deployment.

Our contributions are threefold:

- **We reveal execution-order variation as a source of learning noise.** A frozen-policy study of 640 outcome-equivalent transcript pairs identifies distinct gradient signals despite unchanged task outcomes, exposing a source of variation beyond reward attribution (Section 3).
- **We propose policy optimization over verified workflow alternatives.** PWPO preserves the expected on-policy gradient while removing variance within verified pairs. Our theory also characterizes when this statistical gain offsets the added training cost (Section 4.4).
- **PWPO improves agent performance under matched rollout budgets.** It achieves the highest mean across all 16 main-table metrics on AppWorld-Atomic, ALFWorld, and WebShop. Relative to G2PO, AppWorld-Atomic task completion improves by 2.32–2.64 percentage points (Tables 1–2).

2 Related Work

Structured credit assignment. Beyond episode-level group comparisons (Shao et al., 2024), GiGPO derives local advantages from repeated states (Feng et al., 2025); HGPO addresses the resulting context mismatch by grouping steps at multiple history depths (He et al., 2026). Graph-based methods further pool state-value estimates across trajectories (G2PO) (Wang et al., 2026a) or share continuations among semantically similar reasoning states (GraphPO) (Zhan et al., 2026). BEACON anchors credit to task milestones and rewards partial progress (Wang et al., 2026b). PWPO complements these approaches to reward attribution by aggregating the policy probabilities of interchangeable execution orders. It retains terminal rewards and assigns credit to the combined probability of a certified transcript pair.

Workflow structure and execution order. Agent Workflow Memory extracts reusable routines to guide subsequent actions (Wang et al., 2025), while BPOP infers latent partial orders from successful traces and compiles them into executable workflows (Li et al., 2026). For probability aggregation, interchangeable execution orders must also preserve outputs and reward. PWPO verifies a local exchange from a complete environment snapshot, preserving per-call outputs, endpoint state, and reward. This local construction supports exact probability aggregation; unverified exchanges remain singletons.

Marginalization and the optimization unit. CAPG and marginal policy gradients reduce variance by integrating over actions that the environment treats as equivalent (Fujita & Maeda, 2018; Eisenach et al., 2019). Graph Diffusion Policy Optimization also motivates learning over trajectories with the same terminal graph, using a biased eager gradient (Liu et al., 2024). For language models, GSPO clips length-normalized sequence likelihood ratios for individual responses (Zheng et al., 2025). PWPO applies the established marginalization principle to certified tool-order pairs: it sums complete transcript probabilities, including suffixes evaluated under their respective histories, before forming the policy ratio. The resulting covariance guarantee concerns exact on-policy gradients; certification and rescoring costs enter a separate budget analysis.

3 PRELIMINARY STUDY

Does outcome equivalence imply learning equivalence? An autoregressive policy conditions on execution history, so outcome-equivalent orders can yield different learning signals. We examine 640 transcript pairs under a frozen policy. Each pair swaps two adjacent independent call–output blocks while preserving their outputs, pair-end state, termination, and reward. Using the recorded 64-dimensional policy scores (gradients of transcript log-probability), we measure the *order-noise fraction* ν: the share of the pair-conditioned gradient second moment attributable to sampling the execution order. Appendix B.5 details the protocol, metrics, and visualization.

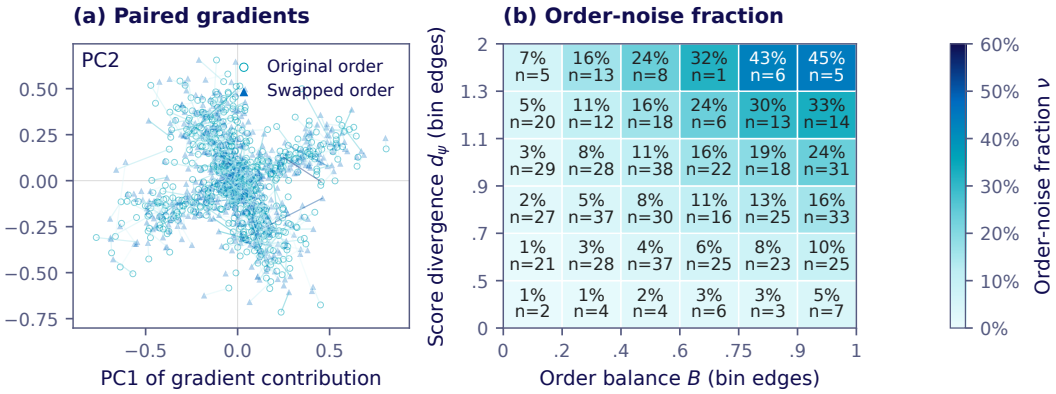


Figure 2: **Equivalent outcomes, different learning signals. (a)** Paired gradient contributions in a common PCA projection. **(b)** Median order-noise fraction ν by order balance B and score divergence d_ψ; cells report medians and counts. Appendix B.5.3 specifies the projection and bins.

Order variation is concentrated. Across pairs, the median absolute log-probability gap is 1.66 nats and the median score cosine is 0.63, revealing distinct policy responses despite equivalent outcomes (Figure 2). The median ν is 8.9%, with a 90th percentile of 23.3%. When both orders are plausible and their scores strongly disagree, the median rises to 32.1%; with low order balance or low score divergence, it is 4.2% (group definitions in Appendix C.7). These observations motivate optimizing the combined probability of equivalent orders (Section 4).

4 METHOD

Overview. Pairwise Workflow Policy Optimization (PWPO) combines the probabilities of a rollout and a certified reordering of two adjacent tool calls (Figure 3). It samples one boundary per rollout and accepts the swap only if it preserves the interaction outcome; otherwise, it keeps the original transcript alone. The boundary and transcript set define a learning event, or *cell*, whose exact probability ratio enters a clipped policy objective. On-policy, averaging over the certified orders preserves the expected gradient while reducing variation due to execution order.

4.1 CERTIFIED LOCAL CELLS

Interaction and likelihood. We assume deterministic tool execution and restorable, complete environment snapshots. A task x specifies the instruction and initial state. Under a fixed token grammar, each policy response is one atomic tool call or a terminal submission. A complete transcript τ records all sampled tokens, tool outputs, and the termination status, including failed episodes and unfinished final responses. Its probability and the learning objective are

$$p_\theta(\tau \mid x) = \prod_{r=1}^{L(\tau)} \pi_\theta(v_r \mid h_r; \mathcal{G}), \qquad J(\theta) = \mathbb{E}_{x,\, \tau \sim p_\theta(\cdot|x)}[R_x(\tau)]. \tag{1}$$

Here v_r is the rth sampled token, h_r its full visible history, and $L(\tau)$ the number of sampled tokens; token probabilities are normalized under the fixed grammar $\mathcal{G}$. Deterministic tool outputs and forced

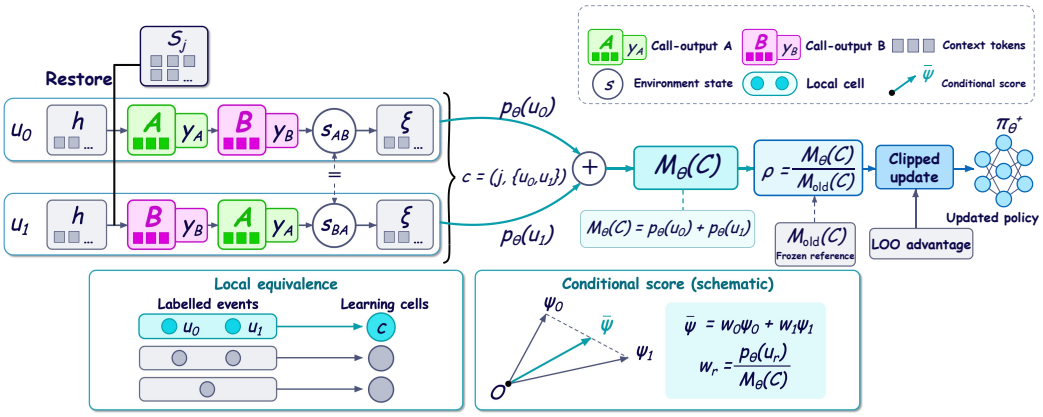


Figure 3: **Overview of PWPO.** Starting from the shared snapshot after prefix h, certification checks both orders of blocks (A, y_A) and (B, y_B) for protocol validity, arguments resolvable from the prefix, matching outputs, and equal endpoint states. Accepted swaps form a labelled cell $c = (j, \{u_0, u_1\})$; rejected swaps remain singletons. Full-transcript probabilities, with suffix ξ rescored under each history, sum to $M_\theta(C)$. The ratio to its frozen behavior-policy value drives a clipped update with leave-one-out (LOO) advantages. Lower insets show cell formation and the probability-weighted score $\bar{\psi}_\theta(c) = \nabla_\theta \log M_\theta(C)$. Task conditioning is omitted.

termination markers contribute no policy factors. Token and call limits, validation, and stopping rules define a finite support independent of θ. The reward $R_x(\tau) = r_x(s_{\text{end}}, d, \eta)$ depends on the final state s_{end}, submission payload d ($\perp$ if absent), and termination status η. Failure statuses receive a fixed task-dependent reward. Appendix A.1.1 gives the complete stopping protocol.

Certifying an adjacent swap. Let $m(\tau)$ count completed tool call–output blocks. If $m(\tau) \geq 2$, draw j uniformly from the $m(\tau)-1$ adjacent boundaries; otherwise, set $j = 0$. Let $S_j\tau$ swap blocks j and $j+1$ without changing their tokens, outputs, or identifiers. The certificate $\kappa_x(\tau, j, s_j)$ receives the full transcript, selected boundary, and snapshot s_j immediately before the pair. Both transcripts must obey the interaction protocol, and both calls' arguments must resolve from the common prefix. Replaying the two orders must satisfy

$$\begin{aligned} T(T(s_j, a_j), a_{j+1}) &= T(T(s_j, a_{j+1}), a_j), \\ Y(s_j, a_j) &= Y(T(s_j, a_{j+1}), a_j), \\ Y(s_j, a_{j+1}) &= Y(T(s_j, a_j), a_{j+1}). \end{aligned} \tag{2}$$

Here a_j, a_{j+1} are the concrete calls, T the state transition, and Y the tool output. Both calls must succeed in each order without intermediate termination. The first equality preserves execution of the unchanged suffix; the other two preserve each call's output. The certificate validates references in both full histories and ensures that the submission and termination status are preserved. Applying these checks to the unordered pair $\{\tau, S_j\tau\}$ makes acceptance symmetric (Appendix A.1.2).

The learning event. Define $F_j\tau = S_j\tau$ on acceptance and $F_j\tau = \tau$ on rejection, with F_0 the identity. Symmetry makes the swap reversible, $F_j^2 = I$, and certification preserves reward, $R_x(F_j\tau) = R_x(\tau)$. The set $C_j(\tau) = \{\tau, F_j\tau\}$ contains at most two distinct transcripts. Because pairs from different boundaries can overlap, we retain the boundary in the *labelled cell* $c = (j, C_j(\tau))$. These cells partition the augmented sample space (τ, j), with probabilities

$$M_\theta(C \mid x) = \sum_{u \in C} p_\theta(u \mid x), \qquad q_\theta(c \mid x) = \nu(j \mid m(C)) M_\theta(C \mid x). \tag{3}$$

Here M_θ is the total probability of C, $m(C)$ its members' common call count, and $\nu(j \mid m) = 1/(m-1)$ for $m \geq 2$, with $\nu(0 \mid m) = 1$ otherwise. Thus $\sum_c q_\theta(c \mid x) = 1$. Rejected boundaries remain singletons without resampling; identical members are counted once.

4.2 Optimizing exact cell probabilities

Advantages and likelihood ratios. For each task x_t in a batch of B tasks, independently sample $N \geq 2$ transcripts from the frozen behavior policy θ_{old}. Construct each cell $c_{ti} = (j_{ti}, C_{ti})$ once and compute the LOO advantage

$$A_{ti} = R_{ti} - \frac{1}{N-1} \sum_{k \neq i} R_{tk}, \qquad R_{ti} = R_{x_t}(\tau_{ti}). \tag{4}$$

The baseline is the mean reward of the other independent rollouts for the same task. Rewards, advantages, cells, and behavior-policy likelihoods stay fixed throughout the update. Advantages are not divided by the reward standard deviation.

The exact cell ratio is $\rho_\theta(c) = q_\theta(c \mid x)/q_{\text{old}}(c \mid x)$. The policy-independent boundary selector cancels, giving $\rho_\theta(c) = M_\theta(C \mid x)/M_{\text{old}}(C \mid x)$. Write $\rho_{ti}(\theta)$ for this ratio on c_{ti}.

To compute it, evaluate each distinct member's full token log-likelihood $\ell_\theta(u \mid x_t) = \log p_\theta(u \mid x_t)$ and form the cell log-mass $\text{LSE}_{u \in C_{ti}} \ell_\theta(u \mid x_t)$, where $\text{LSE}(z_1, \ldots, z_n) = \log \sum_r e^{z_r}$. Subtract the old log-mass and exponentiate to obtain ρ_{ti}. Even the unchanged suffix must be rescored: reordering changes the policy's visible history despite preserving the environment state (Appendix A.1.3).

Policy update. PWPO maximizes the cell-level clipped objective (Schulman et al., 2017),

$$\mathcal{L}(\theta) = \frac{1}{BN} \sum_{t=1}^{B} \sum_{i=1}^{N} \min\{\rho_{ti}(\theta) A_{ti},\ \text{clip}(\rho_{ti}(\theta), 1-\epsilon, 1+\epsilon)\, A_{ti}\}\,, \tag{5}$$

where $\epsilon \in (0, 1)$ is the clipping width. Gradients pass through the current log-sum-exp; cells and old log-masses are cached for a fixed number of update epochs. Every sampled rollout contributes separately, even if several map to the same cell. Singletons recover the trajectory-level objective exactly. Algorithm 1 gives the full procedure.

4.3 Conditional scores and variance reduction

The gradient of a cell's log-probability is a probability-weighted average of its members' score vectors. For a fixed task x, define the full-transcript *score vector* $\psi_\theta(u) = \nabla_\theta \log p_\theta(u \mid x)$ and the conditional member probability $w_\theta(u \mid c) = p_\theta(u \mid x)/M_\theta(C \mid x)$ for $c = (j, C)$. Differentiating the cell log-mass gives

$$\bar{\psi}_\theta(c) := \nabla_\theta \log M_\theta(C \mid x) = \sum_{u \in C} w_\theta(u \mid c) \psi_\theta(u) = \mathbb{E}[\psi_\theta(\tau) \mid c, x]. \tag{6}$$

The boundary selector cancels because cell members have the same call count. Averaging removes their score differences: formally, it is a weighted orthogonal projection of transcript-indexed score functions onto functions constant within each labelled cell (Appendix D.1.1).

Proposition 1 (On-policy mean and covariance)**.** *Fix a task x and a rollout-independent baseline $\beta(x)$. Let $R_x(c)$ be the members' common reward and $A(c) = R_x(c) - \beta(x)$. Define $g_{\text{seq}} = A(c)\psi_\theta(\tau)$ and $g_{\text{pair}} = A(c)\bar{\psi}_\theta(c)$. If probabilities are differentiable with common positive support and the estimators have finite second moments, then*

$$\begin{aligned} \mathbb{E}[g_{\text{pair}} \mid x] &= \mathbb{E}[g_{\text{seq}} \mid x] = \nabla_\theta J_x(\theta), \\ \text{Cov}(g_{\text{seq}} \mid x) - \text{Cov}(g_{\text{pair}} \mid x) &= \mathbb{E}_{c|x}\left[A(c)^2 w_0 w_1 \Delta\psi_C \Delta\psi_C^\top\right] \succeq 0. \end{aligned} \tag{7}$$

Here J_x is the expected task reward. For $C = \{u_0, u_1\}$, $w_r = w_\theta(u_r \mid c)$ and $\Delta\psi_C = \psi_\theta(u_0) - \psi_\theta(u_1)$; singleton cells contribute zero.

Each two-member cell removes a conditional covariance term of rank at most one. This local Rao–Blackwellization follows the marginalization principle used in marginal policy gradients and CAPG (Eisenach et al., 2019; Fujita & Maeda, 2018). Appendix D.1.2 proves Proposition 1; Appendix D.1.3 extends it to batches with shared LOO advantages. The covariance guarantee holds at $\theta = \theta_{\text{old}}$, where clipping is inactive and Equation (5) has this conditional-mean gradient.

PWPO attempts certification once per rollout. Each two-member cell adds one full-transcript likelihood per policy evaluation compared with a trajectory-level update. The budget analysis below accounts for replay and scoring costs. Deployment uses the original sequential policy interface.

4.4 BUDGET-CONSTRAINED STATISTICAL EFFICIENCY

We quantify when variance reduction offsets the added cost. Fix $\theta = \theta_{\text{old}}$ and assume exact certification and the regularity and moment conditions in Appendix D.2. Sample independent *task blocks*, each with a task x and $N \geq 2$ rollouts sampled independently given x. Each block returns $G_{e,N} = N^{-1} \sum_i A_i S_{e,i}$ with the LOO advantages in Equation (4). The trajectory estimator ($e = \text{s}$) uses $S_{\text{s}} = \psi_\theta(\tau)$; PWPO ($e = \text{p}$) uses $S_{\text{p}} = \bar{\psi}_\theta(c)$. For $g = \nabla J(\theta)$, define the block mean squared error (MSE) $V_e(N) = \mathbb{E}\|G_{e,N} - g\|^2$ and its reduction $\Delta_N = V_{\text{s}}(N) - V_{\text{p}}(N)$.

Theorem 1 (When PWPO saves computation)**.** *Let $c_e(N) > 0$ be the expected total block cost, including rollout collection, certification, replay, likelihood evaluation, and differentiation. For an expected budget $\mathcal{B}$, choose $K_e = \lfloor \mathcal{B}/c_e(N) \rfloor \geq 1$ blocks in advance and average their estimates to obtain $\widehat{g}_e$. Both estimators are unbiased, $V_{\text{p}}(N) \leq V_{\text{s}}(N)$, and, for $V_{\text{s}}(N) > 0$,*

$$\begin{aligned} \mathcal{R}_e(\mathcal{B}, N) &:= \mathbb{E}\|\widehat{g}_e - g\|^2 = \frac{V_e(N)}{K_e}, \\ \mathcal{R}_{\text{p}}(\mathcal{B}, N) < \mathcal{R}_{\text{s}}(\mathcal{B}, N) \quad &\Longleftrightarrow \quad \frac{\Delta_N}{V_{\text{s}}(N)} > 1 - \frac{K_{\text{p}}}{K_{\text{s}}}. \end{aligned} \tag{8}$$

As the budget grows with N fixed, $\mathcal{B}\mathcal{R}_e(\mathcal{B}, N) \to Q_e(N) := c_e(N) V_e(N)$. If PWPO adds relative cost $\kappa_N \geq 0$, with $c_{\text{p}}(N) = (1 + \kappa_N) c_{\text{s}}(N)$, then

$$Q_{\text{p}}(N) < Q_{\text{s}}(N) \quad \Longleftrightarrow \quad \frac{\Delta_N}{V_{\text{s}}(N)} > \frac{\kappa_N}{1 + \kappa_N}. \tag{9}$$

This condition guarantees smaller MSE for all sufficiently large expected budgets at the same N.

Proof sketch. An order-two U-statistic decomposition gives the exact covariance with shared LOO baselines. Conditioning on labelled cells removes a positive semidefinite covariance term. Averaging independent blocks gives MSE V_e/K_e; comparing these errors and taking the large-budget limit at fixed N gives the two thresholds. Appendix D.3–D.4 provides the variance decomposition and full proofs. □

For fixed N, PWPO needs no more original rollouts to reach a given MSE; computational savings depend on the variance reduction relative to the added cost. Appendix D.4 gives exact sample counts and comparisons with N optimized separately for each method. These guarantees apply to gradient estimation at a fixed on-policy checkpoint.

5 EXPERIMENTS

Building on the order-noise diagnosis in Section 3, we organize the evaluation around four questions:

RQ1. Does workflow equivalence improve task success? (Section 5.2)

RQ2. Which parts of the learning mechanism matter? (Section 5.3)

RQ3. When does variance reduction justify the extra cost? (Section 5.4)

RQ4. Where is removable order noise concentrated? (Section 5.5)

5.1 EXPERIMENTAL SETUP

Benchmarks and metrics. The benchmarks are AppWorld-Atomic (Trivedi et al., 2024), ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and supplementary ScienceWorld (Wang et al., 2022). Metrics include AppWorld task/scenario goal completion, ALFWorld success, and WebShop/ScienceWorld score and success, supplemented by certification coverage and cost-adjusted gradient variance (Appendix B.3).

Comparison protocol. Baselines include PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), GiGPO (Feng et al., 2025), HGPO (He et al., 2026), and G2PO (Wang et al., 2026a). Comparisons share checkpoints, splits, atomic interfaces, terminal rewards, and rollout budgets; the compute-matched comparison uses AppWorld-Atomic at 7B. BEACON (Wang et al., 2026b) uses a separate feedback-matched comparison; additional controls and interface distinctions appear in Appendix B.2.

Backbones and implementation. Backbones are Qwen2.5-Instruct at 1.5B, 7B, and 14B parameters (Qwen Team, 2024). The default configuration uses $N = 8$ rollouts per task, actor learning rate 10^{-6}, and clipping width $\epsilon = 0.2$; hardware, optimization, and tuning details appear in Appendix B.4.

5.2 RQ1: Does workflow equivalence improve task success?

Performance across tasks and model scales. With 19,200 original rollouts per method, PWPO achieves the highest mean in all 16 metrics in Tables 1–2. Relative to G2PO, AppWorld-Atomic task goal completion improves by 2.32–2.64 percentage points across splits and scales; ALFWorld success improves by 1.43–3.73 points, and WebShop success by 2.48/2.00 points at 1.5B/7B. The gains span software tools, household tasks, and shopping under the shared terminal-reward protocol.

Table 1: **AppWorld-Atomic results.** Task/scenario goal completion (TGC/SGC; %, ↑) on Test-N and Test-C. All methods share the 100-call limit; trained methods use 19,200 rollouts. Results are mean ± standard deviation across seeds; bold denotes the highest mean in each column.

	Qwen2.5-7B-Instruct				Qwen2.5-14B-Instruct			
	Test-N		Test-C		Test-N		Test-C	
Method	TGC	SGC	TGC	SGC	TGC	SGC	TGC	SGC
Initial checkpoint	$9.72_{\pm 0.34}$	$5.95_{\pm 1.03}$	$7.19_{\pm 0.24}$	$3.12_{\pm 0.42}$	$13.29_{\pm 0.34}$	$9.52_{\pm 1.03}$	$9.83_{\pm 0.24}$	$5.28_{\pm 0.42}$
PPO	$18.45_{\pm 1.19}$	$13.10_{\pm 2.06}$	$13.91_{\pm 0.63}$	$10.31_{\pm 0.83}$	$22.62_{\pm 0.60}$	$15.48_{\pm 1.03}$	$18.86_{\pm 0.37}$	$14.63_{\pm 0.42}$
RLOO	$19.64_{\pm 1.19}$	$14.88_{\pm 2.73}$	$14.87_{\pm 0.63}$	$11.03_{\pm 1.10}$	$24.40_{\pm 0.60}$	$17.26_{\pm 1.03}$	$20.06_{\pm 0.37}$	$15.59_{\pm 0.42}$
GRPO	$20.63_{\pm 0.91}$	$15.48_{\pm 2.06}$	$15.59_{\pm 0.72}$	$11.75_{\pm 1.10}$	$25.20_{\pm 0.69}$	$18.45_{\pm 1.03}$	$20.86_{\pm 0.48}$	$16.31_{\pm 0.42}$
GiGPO	$22.82_{\pm 0.69}$	$16.67_{\pm 1.03}$	$17.75_{\pm 0.72}$	$13.43_{\pm 0.83}$	$27.18_{\pm 0.34}$	$20.24_{\pm 1.03}$	$22.38_{\pm 0.50}$	$17.51_{\pm 0.42}$
HGPO	$23.41_{\pm 0.91}$	$17.26_{\pm 1.03}$	$18.47_{\pm 0.72}$	$13.91_{\pm 0.83}$	$27.98_{\pm 0.60}$	$20.83_{\pm 1.03}$	$23.10_{\pm 0.50}$	$17.99_{\pm 0.72}$
G2PO	$24.80_{\pm 0.91}$	$18.45_{\pm 1.03}$	$20.38_{\pm 0.72}$	$14.87_{\pm 0.83}$	$29.37_{\pm 0.69}$	$22.02_{\pm 1.03}$	$24.86_{\pm 0.37}$	$19.18_{\pm 0.42}$
Seq-LOO	$24.01_{\pm 0.91}$	$17.26_{\pm 1.03}$	$19.42_{\pm 0.72}$	$14.15_{\pm 0.42}$	$28.57_{\pm 0.60}$	$20.83_{\pm 1.03}$	$23.82_{\pm 0.50}$	$18.47_{\pm 0.42}$
PWPO (ours)	$\mathbf{27.38_{\pm 1.19}}$	$\mathbf{20.83_{\pm 1.03}}$	$\mathbf{23.02_{\pm 0.72}}$	$\mathbf{17.03_{\pm 0.42}}$	$\mathbf{31.75_{\pm 0.69}}$	$\mathbf{24.40_{\pm 1.03}}$	$\mathbf{27.18_{\pm 0.50}}$	$\mathbf{21.82_{\pm 0.42}}$

Table 2: **ALFWorld and WebShop results.** ALFWorld seen/unseen success and WebShop score/success (%, ↑). All methods share the 50/15-action limits; trained methods use 19,200 rollouts. Results are mean ± standard deviation across seeds; bold denotes the highest mean in each column.

	Qwen2.5-1.5B-Instruct				Qwen2.5-7B-Instruct			
	ALFWorld		WebShop		ALFWorld		WebShop	
Method	Seen	Unseen	Score	Succ.	Seen	Unseen	Score	Succ.
Initial checkpoint	$43.29_{\pm 0.81}$	$37.16_{\pm 0.97}$	$55.40_{\pm 0.82}$	$40.28_{\pm 0.54}$	$67.14_{\pm 0.71}$	$60.45_{\pm 0.75}$	$68.42_{\pm 0.74}$	$57.87_{\pm 0.50}$
PPO	$64.29_{\pm 1.13}$	$58.96_{\pm 1.18}$	$68.20_{\pm 1.36}$	$52.64_{\pm 0.67}$	$81.19_{\pm 1.09}$	$76.87_{\pm 0.75}$	$81.55_{\pm 1.80}$	$68.80_{\pm 0.60}$
RLOO	$62.14_{\pm 1.13}$	$55.97_{\pm 1.18}$	$67.05_{\pm 1.21}$	$51.04_{\pm 0.48}$	$76.19_{\pm 1.09}$	$71.89_{\pm 1.14}$	$80.36_{\pm 1.88}$	$65.73_{\pm 0.50}$
GRPO	$66.43_{\pm 1.13}$	$60.45_{\pm 1.18}$	$69.55_{\pm 1.10}$	$54.12_{\pm 0.41}$	$78.33_{\pm 1.09}$	$74.38_{\pm 1.14}$	$79.71_{\pm 1.61}$	$66.13_{\pm 0.50}$
GiGPO	$74.29_{\pm 1.13}$	$70.15_{\pm 1.18}$	$75.18_{\pm 0.92}$	$60.20_{\pm 0.55}$	$93.57_{\pm 0.71}$	$92.54_{\pm 0.75}$	$86.02_{\pm 1.27}$	$74.80_{\pm 0.60}$
HGPO	$76.43_{\pm 1.13}$	$72.39_{\pm 1.18}$	$76.04_{\pm 0.81}$	$61.68_{\pm 0.48}$	$95.71_{\pm 0.71}$	$93.53_{\pm 0.86}$	$86.63_{\pm 1.15}$	$75.27_{\pm 0.50}$
G2PO	$79.29_{\pm 1.13}$	$75.37_{\pm 1.18}$	$78.68_{\pm 0.74}$	$64.04_{\pm 0.48}$	$97.38_{\pm 0.41}$	$96.27_{\pm 0.75}$	$89.62_{\pm 0.91}$	$78.27_{\pm 0.61}$
Seq-LOO	$74.29_{\pm 1.13}$	$69.40_{\pm 1.18}$	$74.72_{\pm 1.03}$	$59.76_{\pm 0.38}$	$90.71_{\pm 0.71}$	$88.56_{\pm 1.14}$	$85.44_{\pm 1.36}$	$73.53_{\pm 0.50}$
PWPO (ours)	$\mathbf{82.86_{\pm 1.13}}$	$\mathbf{79.10_{\pm 1.18}}$	$\mathbf{80.86_{\pm 0.65}}$	$\mathbf{66.52_{\pm 0.54}}$	$\mathbf{98.81_{\pm 0.41}}$	$\mathbf{97.76_{\pm 0.75}}$	$\mathbf{91.18_{\pm 0.78}}$	$\mathbf{80.27_{\pm 0.42}}$

Longer workflows and feedback controls. At 30 actions, PWPO improves success over G2PO by 2.24/1.47 points on WebShop and 2.84/2.28 on ScienceWorld at 1.5B/7B (Table 3). BEACON reaches higher ScienceWorld scores with additional milestone feedback, shared by its GRPO and Seq-LOO controls (Table 6).

Table 3: **Success on 30-action workflows.** Mean ± standard deviation (%, ↑) with terminal rewards and 19,200 rollouts. Full scores and additional feedback controls appear in Tables 5–6; bold marks the highest mean shown.

	WebShop		ScienceWorld	
Method	1.5B	7B	1.5B	7B
GiGPO	$63.16_{\pm 0.52}$	$77.67_{\pm 0.42}$	$25.86_{\pm 1.12}$	$53.55_{\pm 0.91}$
HGPO	$64.88_{\pm 0.48}$	$79.27_{\pm 0.58}$	$28.74_{\pm 1.06}$	$55.43_{\pm 0.86}$
G2PO	$67.00_{\pm 0.55}$	$80.53_{\pm 0.31}$	$32.64_{\pm 1.14}$	$57.36_{\pm 0.83}$
Seq-LOO	$62.00_{\pm 0.45}$	$76.53_{\pm 0.50}$	$31.72_{\pm 1.08}$	$56.04_{\pm 0.88}$
PWPO (ours)	$\mathbf{69.24}_{\pm \mathbf{0.61}}$	$\mathbf{82.00}_{\pm \mathbf{0.40}}$	$\mathbf{35.48}_{\pm \mathbf{0.99}}$	$\mathbf{59.64}_{\pm \mathbf{0.75}}$

Takeaway 1. Reward-preserving structure in existing executions can supply useful training information without requiring a richer reward signal.

5.3 RQ2: Which parts of the learning mechanism matter?

Pair aggregation and clipping. Figure 4 compares sequence-level training, uniform pair weighting, and frozen-weight memberwise clipping. PWPO improves completion over Seq-LOO by 2.88–3.60 points and over Uniform-Pair by 1.20–2.38 points, with all paired 95% intervals above zero (Table 8). The 0.48–0.99-point gains over Frozen-weight Pair have intervals containing zero. The clearest evidence is for probability-weighted aggregation.

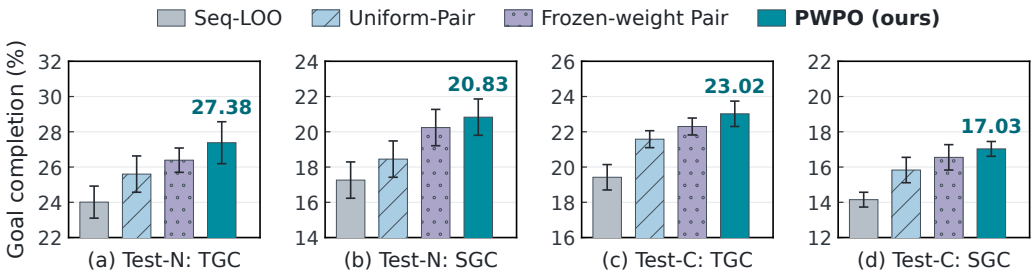


Figure 4: **PWPO leads in mean completion.** AppWorld-Atomic/7B, 100 calls, 19,200 rollouts. Bars: means; whiskers: standard deviations; labels: PWPO means. Truncated y-axes span 10 points. Full results, costs, and paired intervals: Table 8.

Why the weights matter. Exact enumeration in the finite tool model separates estimator bias from variance (Table 10). PWPO and Frozen-weight Pair retain zero squared bias at the reported precision and reduce the variance ratio to 0.742, whereas Uniform-Pair introduces a squared-bias ratio of 0.034. Disabling pair acceptance recovers Seq-LOO for every control, validating the singleton fallback in this finite model.

Where certification is available. The replay audit finds accepted pairs in 23.9–41.6% of original rollouts on the three main benchmarks and 8.3–9.9% on ScienceWorld (Table 9). No restore/replay faults or accepted-pair violations were observed; full task results include rejected and singleton records.

Takeaway 2. Execution equivalence determines which records may share credit; their policy probabilities determine how to share it without shifting the learning target.

5.4 RQ3: When does variance reduction justify the extra cost?

Task performance under a compute budget. At matched rollout counts, PWPO uses 451.5 GPU-hours versus 382.6 for Seq-LOO (Table 8). With compute matched on AppWorld-Atomic/7B, PWPO exceeds G2PO by 2.19/2.16 task-completion points on Test-N/Test-C at 381.7 versus 383.1 GPU-hours (Table 7(a)). Tool calls including replay rise from 169.6k to 213.9k, so the GPU-matched gain costs more interactions.

Variance reduction after accounting for cost. At the fixed AppWorld checkpoint, PWPO's variance ratio of 0.77 and cost ratio of 1.18 yield a cost–variance ratio of 0.91 relative to Seq-LOO (paired 95% interval $[0.83, 0.99]$; Table 11). Uniform-Pair reduces variance but has a cost–variance ratio of 1.03 ($[0.97, 1.10]$). Under the prespecified finite budget, averaging 32 PWPO blocks versus 38 Seq-LOO blocks gives a variance ratio of 0.91 ($[0.84, 1.00]$). Frozen-weight Pair matches PWPO at the reported precision.

Takeaway 3. Certification pays off when the removed gradient noise offsets its total cost. Its value depends on whether GPU time or tool interaction is the binding resource.

5.5 RQ4: Where is removable order noise concentrated?

We analyze 1,600 certified AppWorld-Atomic pairs from the fixed 7B policy ($N = 8$; Table 7(b)). For each pair, we decompose removable conditional covariance across 28 depth bins and normalize by its total mass. Figure 5 maps median layer shares against workflow position $p = j/(m-1)$: j indexes the exchanged boundary and m counts completed tool calls (Appendix B.6).

The high-share ridge moves from depth bins 11–13 at $p < 0.24$ to bins 18–19 at $p \geq 0.48$ and remains elevated through the final workflow positions. Bins 15–28 account for 73.91% of the within-pair mass on average, compared with 7.34% in bins 1–9. The dominant-bin density is highest at bin 18 and $p \in [0.56, 0.64)$.

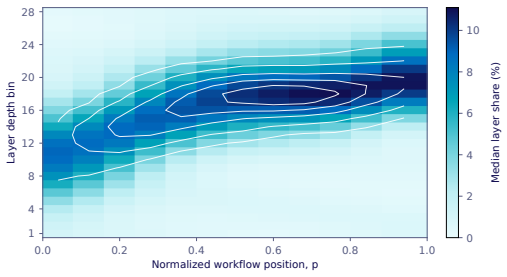


Figure 5: **Order-noise concentration shifts to deeper layers as workflows progress.** Color shows the smoothed median within-pair share of removable variance; white contours show the density of each pair's dominant depth bin.

Concentration over workflow progress. Positions $p \in [0.32, 0.72)$ contain 74.36% of the total raw removable mass, weighting pairs by their covariance trace (Table 12). The smoothed median layer share instead peaks at 11.03% in bin 18 and $p \in [0.72, 0.80)$. This within-pair share measures relative depth concentration, whereas the dominant-bin density counts where pairs attain their maxima.

Takeaway 4. Conditioning on workflow progress reveals distinct depth profiles of order sensitivity that averaging across execution positions can obscure.

6 Conclusion

Execution-order variation creates learning noise even when tool interactions preserve task outcomes. PWPO uses verified workflow equivalence to construct an unbiased on-policy gradient estimator and gives explicit conditions for variance reduction to offset computational cost. Controlled experiments show improved task performance under matched rollout budgets and gains at comparable GPU time on AppWorld-Atomic. These results motivate accounting for execution equivalence alongside reward attribution. Extending PWPO to larger sets of verified workflow alternatives requires tractable aggregation that preserves its statistical guarantees.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and document formatting, to check mathematical proofs, and to improve the organization and exposition of the Method section. The authors take full responsibility for all textual content and experimental results reported in this paper.

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. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Carson Eisenach, Haichuan Yang, Ji Liu, and Han Liu. Marginal policy gradients: A unified family of estimators for bounded action spaces with applications. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=HkgqFiAcFm.

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, 2025. doi: 10.52202/085713-1544. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Yasuhiro Fujita and Shin-ichi Maeda. Clipped action policy gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1597–1606, 2018. URL https://proceedings.mlr.press/v80/fujita18a.html.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming, 2013. URL https://arxiv.org/abs/1309.5549.

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. URL https://openreview.net/forum?id=T8Dev99qnz.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948. doi: 10.1214/aoms/1177730196. URL https://doi.org/10.1214/aoms/1177730196.

Dongqing Li, Zheqiao Cheng, Geoff K. Nicholls, and Quyu Kong. De-linearizing agent traces: Bayesian inference of latent partial orders for efficient execution, 2026. URL https://arxiv.org/abs/2602.02806.

Yijing Liu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Wei Chen. Graph diffusion policy optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/124256ed80af5d4bf4c4de17b66c4298-Abstract-Conference.html.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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, 2024. URL https://arxiv.org/abs/2402.03300.

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. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

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. doi: 10.18653/v1/2022.emnlp-main.775. URL https://aclanthology.org/2022.emnlp-main.775/.

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, 2026a. URL https://arxiv.org/abs/2606.22995.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents. In *International Conference on Machine Learning*, 2026b. URL https://icml.cc/virtual/2026/poster/65128.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

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, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Yuliang Zhan, Xinyu Tang, Jian Li, Dandan Zheng, Weilong Chai, Jingdong Chen, Jun Zhou, Ge Wu, Wenyue Tang, and Hao Sun. GraphPO: Graph-based policy optimization for reasoning models, 2026. URL https://arxiv.org/abs/2606.18954.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Appendix Contents

Pairwise Workflow Policy Optimization for Tool-Using Language Agents Page

A Technical Details . . . 13
A.1 Interaction protocol and local certification . . . 13
A.2 Training procedure and execution contract . . . 14

B Experimental Details . . . 16
B.1 Benchmarks, splits, and evaluation interfaces . . . 16
B.2 Baselines and fairness controls . . . 16
B.3 Metrics, budgets, and statistical analysis . . . 17
B.4 Backbones, optimization, and PWPO configuration . . . 18
B.5 Preliminary study: protocol and diagnostics . . . 19
B.6 Layerwise order-noise diagnostic . . . 21

C Additional Experimental Results . . . 22
C.1 Longer-horizon WebShop . . . 22
C.2 ScienceWorld and feedback controls . . . 22
C.3 Compute-matched performance and estimator diagnostics . . . 23
C.4 End-task mechanism comparison . . . 24
C.5 Certification and replay audit . . . 25
C.6 Finite-gradient and cost diagnostics . . . 25
C.7 Preliminary study: summary statistics . . . 26
C.8 Layerwise order-noise statistics . . . 26

D Proofs . . . 27
D.1 Projection and estimator guarantees . . . 27
D.2 Probability space and conditional score moments . . . 28
D.3 Exact LOO decomposition and variance reduction . . . 29
D.4 Finite budgets and exact sample counts . . . 31
D.5 Optimal rollout width under affine expected costs . . . 34
D.6 Sharp coverage-dependent variance envelope . . . 35
D.7 Expected budgets versus deterministic hard caps . . . 36
D.8 Finite-compute optimization with fresh on-policy updates . . . 37

A TECHNICAL DETAILS

A.1 INTERACTION PROTOCOL AND LOCAL CERTIFICATION

A.1.1 TOKEN ACCOUNTING, TERMINATION, AND SUPPORT

The policy interface uses a fixed serialization for two response types: a single named tool with its argument fields, and a terminal submission with its payload. All delimiters are sampled tokens. A completed tool response is validated and executed before the next response is generated. All model-generated text contributes to the likelihood. A transcript retains the exact sampled tokens, observed outputs, and deterministic termination record in their original serialization.

Before collection, fix a total sampled-token limit, a tool-invocation limit, a grammar, and a deterministic validator. At each token position, the policy distribution is normalized over the grammar-allowed tokens. Every admitted token has positive probability. Once a response is complete, process it first: an invalid response terminates with status `invalid`; a valid submission terminates with status `submit`; a valid tool invocation is executed and its output is recorded, with a tool failure terminating as `tool_error`. If no such termination occurs, reaching either limit terminates as `limit`. The token limit also terminates an unfinished response, retaining its partial token sequence. This precedence applies identically during collection and certification.

The submission payload d is present only for status `submit`; otherwise $d = \perp$. The statuses `invalid`, `tool_error`, and `limit` receive the preset failure reward. All these episodes remain in the rollout distribution. A sampled submission contributes its complete token likelihood. An incomplete response contributes the likelihood of exactly its sampled prefix. The externally emitted status marker contributes a factor of one in every case. Completed action–output blocks, including a returned tool-error block, count toward $m(\tau)$; an unexecuted invalid response and a partial final response do not.

For fixed x, this protocol is a finite probability tree whose stochastic edges are token choices and whose other edges are deterministic. Its leaves are the complete interaction records $\mathcal{T}_x$. Normalized token transitions and forced termination at the fixed limits imply $\sum_{\tau \in \mathcal{T}_x} p_\theta(\tau \mid x) = 1$. The grammar, environment, and stopping rules are independent of θ, giving a common support; positivity gives $p_\theta(\tau \mid x) > 0$ on that support. Equation (1) is the product along each root-to-leaf path, including paths ending before a submission.

A.1.2 SYMMETRIC CERTIFICATION AND CLOSURE

The certificate input is (x, τ, j, s_j), with the complete transcript τ and a restorable environment snapshot s_j. The transcript supplies the visible prefix, action tokens, outputs, message identifiers, reference scopes, suffix, and termination record for checking history-dependent validity. The snapshot supplies every environment variable affecting subsequent execution and reward, including mutable stores, counters, and termination flags.

For $j > 0$, form $u = S_j\tau$ by exchanging the two complete action–output blocks. Apply the following deterministic checks to the unordered pair $\{\tau, u\}$:

1. Both records are admitted by the fixed grammar, validator, reference rules, and stopping protocol. The same tokens remain attached to the same call and output, with their original identifiers. Both calls' arguments resolve from the common prefix to the same concrete values in either order. A reference created by the other call fails this check.
2. Restore two copies of s_j and execute both orders. Both calls must complete successfully in each order, without an intermediate termination. Match the returned output to its associated call, check it against the recorded output, and compare the complete endpoint states using Equation (2).

Acceptance requires every check and is independent of rewards and policy likelihoods. Ordinary semantic rejection sets $F_j\tau = \tau$, retaining the selected boundary. A missing required snapshot or inconsistent replay of the recorded order is an infrastructure fault, handled as specified in Appendix A.2.1.

The common prefix and its snapshot are identical for τ and u. Exchanging their roles leaves the unordered pair, admissibility checks, and two-order replay conditions unchanged. Hence $\kappa_x(\tau, j, s_j) = \kappa_x(u, j, s_j)$. Since $S_j^2 = I$, an accepted exchange is reversed by the same rule, while a rejected exchange is the identity. Therefore $F_j^2 = I$. When the exchanged blocks are identical, set formation removes the duplicate and produces a singleton.

For an accepted pair, endpoint equality and deterministic execution preserve the entire fixed suffix, including its outputs. The complete-record checks preserve reference resolution, stopping behavior, and termination status. The final state, submission payload, and status are therefore equal, giving $R_x(F_j\tau) = R_x(\tau)$. Per-call output checks are essential: a write followed by a read can end in the same state as the reverse order while producing a different read output.

A.1.3 Full-transcript likelihoods after an exchange

Write accepted records as $u_0 = h\, a_j y_j\, a_{j+1} y_{j+1}\, \xi$ and $u_1 = h\, a_{j+1} y_{j+1}\, a_j y_j\, \xi$, where h is the common visible prefix and ξ is the unchanged suffix, including its terminal record. For a serialized action a, let $\pi_\theta(a \mid h)$ denote the product of its sampled-token probabilities. Then

$$\frac{p_\theta(u_1 \mid x)}{p_\theta(u_0 \mid x)} = \frac{\pi_\theta(a_{j+1} \mid h)\pi_\theta(a_j \mid h, a_{j+1}, y_{j+1}) p_\theta(\xi \mid h, a_{j+1}, y_{j+1}, a_j, y_j)}{\pi_\theta(a_j \mid h)\pi_\theta(a_{j+1} \mid h, a_j, y_j) p_\theta(\xi \mid h, a_j, y_j, a_{j+1}, y_{j+1})}. \tag{A.1}$$

Here the conditional suffix likelihood includes every remaining sampled token, including an incomplete final response when present; deterministic observations and termination markers contribute factors of one. The common prefix cancels from this within-cell ratio, while the suffix factors depend on the reordered policy histories. The prefix remains part of the cell mass and its gradient.

A.2 Training procedure and execution contract

Algorithm 1 Pairwise Workflow Policy Optimization

Require: Initial policy θ; task distribution; deterministic snapshot environment;
fixed grammar, limits, stopping rules, reward, and certificate;
batch size B, rollouts $N \geq 2$, epochs U, clip width ϵ, optimizer, budget.

while the training budget remains **do**
  Freeze $\theta_{\text{old}} \leftarrow \theta$ and sample tasks $x_{1:B}$.
  for each task x_t **do**
    Independently collect N complete records under the fixed protocol.
    Retain all outcomes and prefix snapshots; compute $A_{t,1:N}$ by (4).
    for each record τ_{ti} **do**
      Draw $j_{ti} \sim \nu(\cdot \mid m(\tau_{ti}))$; initialize $C_{ti} \leftarrow \{\tau_{ti}\}$.
      if $j_{ti} > 0$ **then**
        Evaluate $\kappa_{x_t}(\tau_{ti}, j_{ti}, s_{j_{ti}})$ as in Appendix A.1.2.
        if the certificate accepts **then**
          Set $C_{ti} \leftarrow \{\tau_{ti}, S_{j_{ti}}\tau_{ti}\}$, removing an identical duplicate.
        end if
      end if
      Store $c_{ti} = (j_{ti}, C_{ti})$ and its behavior-policy log-mass.
    end for
  end for
  for $e = 1, \ldots, U$ **do**
    Evaluate full token log-likelihoods for every distinct member of each C_{ti}.
    Compute current log-masses by log-sum-exp and the exact ratios ρ_{ti}.
    Take an optimizer ascent step on Equation (5).
  end for
end while
return π_θ, using sequential tool calls at inference.

A.2.1 LIKELIHOOD EVALUATION AND FAULT HANDLING

Compute each log-mass by the stable identity $\mathrm{LSE}(\ell) = \ell_{\max} + \log \sum_u \exp(\ell_u - \ell_{\max})$. Apply automatic differentiation directly to this expression for the current policy; cache the behavior-policy value without gradients. Common-prefix computation can be shared only while retaining its full current-policy gradient contribution. Every changed-history suffix is evaluated again. Each two-member cell requires two full likelihoods; a singleton requires one.

The statistical analysis assumes correct snapshots, deterministic replay, finite likelihoods, and retention of all collected data. Ordinary certificate rejection retains the sampled record as a singleton. Infrastructure faults, including missing snapshots, inconsistent recorded-order replay, and nonfinite likelihoods, halt training before the affected update. Recovery retains the original batch, boundary draws, behavior-policy parameters, and current optimizer state, then recomputes that update under the same certificate and stopping rules. Training resumes only after this execution contract is restored.

Cells remain fixed throughout all U epochs. Repeated sampled records keep their original multiplicity and leave-one-out advantages. Equal rewards within a task give zero advantages while retaining that task's data and computational cost. Total training cost includes snapshot storage, restoration, two-order replay, behavior-policy likelihood evaluation, and current-policy forward and backward computation. The budget analysis in Section 4.4 combines these costs with the covariance reduction.

A.2.2 ATOMIC ACTION INTERFACE

The action unit is one serialized API invocation, with its returned observation exposed before the next response. AppWorld also supports interactive code generation with complex control flow (Trivedi et al., 2024), where a generated program can contain multiple invocations. Its policy probability belongs to the generated program as a whole. Applying PWPO at the API-call level therefore requires an explicit action-level policy and a complete snapshot contract.

The method requires all execution and reward-relevant state to be restorable, including any clocks, external effects, and sources of randomness. The reward depends only on the final state, submission, and termination status. Each cell aggregates at most one certified adjacent alternative. When every cell is a singleton, PWPO reduces to the underlying trajectory-level update and still incurs certification overhead.

B EXPERIMENTAL DETAILS

The comparisons use common benchmark interfaces, reward protocols, and resource budgets. AppWorld's atomic interface requires validated complete-state snapshots.

B.1 BENCHMARKS, SPLITS, AND EVALUATION INTERFACES

AppWorld. AppWorld (Trivedi et al., 2024) is the primary software-tool benchmark because its tasks combine multiple API calls and persistent application state. We retain the official Train/Dev/Test-N/Test-C task split (105/60/168/417 tasks). Training uses Train and model selection uses Dev. Test-N and Test-C are evaluated separately, with all tasks included regardless of certification success. Task goal completion (TGC) requires passing the official task evaluator, including constraints against unintended state changes; scenario goal completion (SGC) requires completing all three task variants within a scenario. Both metrics are percentages; paired AppWorld comparisons resample scenarios as described below. The *AppWorld-Atomic* interface exposes one documented API invocation per generated response, followed by its observation, or a terminal submission. Native AppWorld accepts generated code with control flow and potentially multiple API calls. We report atomic-interface results separately from native code-turn scores. The atomic-call cap is 100; G2PO uses a limit of 30 code turns (Wang et al., 2026a).

ALFWorld and WebShop. These are the common benchmark pair in GiGPO, HGPO, and G2PO (Feng et al., 2025; He et al., 2026; Wang et al., 2026a). For ALFWorld (Shridhar et al., 2021), we use the text environment, all six task types, and the official `train`, `valid_seen`, and `valid_unseen` identifiers. A fixed 10% split of training task families supplies the development data used for tuning. We report seen/unseen success separately, with a 50-action limit. For WebShop (Yao et al., 2022), we retain the full product catalog and the official baseline split: instruction indices 0–499 for test, 500–1499 for development, and the remaining indices for training. We report mean native purchase reward multiplied by 100 (Score) and the percentage of tasks with reward one (Success). The main action limit is 15, following GiGPO/G2PO; a separate 30-action comparison addresses HGPO's longer horizon. We retain each environment's native action and observation content inside the common serialization and expose no hidden state to the policy.

ScienceWorld. ScienceWorld (Wang et al., 2022), also used by BEACON (Wang et al., 2026b), extends coverage to all 30 scientific task types with official train/development/test variations. We report the native normalized progress score and full-completion rate, using the same task aggregation for every method. The action limit is 30, matching BEACON. Simulated time, evolving objects, and progress counters must be included in snapshots, so certification may reject most exchanges.

Snapshot and replay audit. Every adapted environment must pass exact restore/replay checks for observations, argument resolution, and all state affecting future validity, transitions, termination, or reward. Certification follows Appendix A.1.2, checking complete-record admissibility, both execution orders, associated outputs, and complete endpoint states. Ordinary rejection retains the sampled record as a singleton. A missing snapshot or inconsistent original-order replay is an infrastructure fault and halts the affected update under Appendix A.2.1. The method requires environments that satisfy this restore/replay contract. All tasks and rollouts are retained regardless of exchangeability or reward. Table 9 records PWPO's audit counts for each benchmark/backbone in the reported model matrix.

B.2 BASELINES AND FAIRNESS CONTROLS

Outcome-reward methods. PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), and GRPO (Shao et al., 2024) are recurring training baselines in the main comparison tables of GiGPO, HGPO, G2PO, and BEACON. We add GiGPO (Feng et al., 2025) for state-conditioned grouping, HGPO (He et al., 2026) for history-conditioned hierarchical grouping, and G2PO (Wang et al., 2026a) for graph-based credit assignment. ALFWorld/WebShop form the shared comparison suite; AppWorld-Atomic uses the same methods after interface adaptation, with G2PO as the closest benchmark-matched recent comparator. We retain each baseline's defining advantage estimator and

clipping granularity. PPO's critic is initialized from the same backbone and charged to its budget. The initial checkpoint rows in Tables 1 and 2 use the identical evaluation interface.

Shared reward and information. The controlled outcome-only track uses terminal reward 10 for full task success and 0 otherwise, including invalid responses, tool errors, and limit termination. No step penalty, progress reward, or order-dependent shaping enters PWPO's primary objective. Partial-credit benchmark scores remain evaluation metrics. All methods receive identical instructions, API documentation, observations, action grammar, stopping rules, context limits, and access to training tasks. No external demonstrations, teacher trajectories, or learned reward models are added. Complete hidden snapshots are accessible only to PWPO's deterministic certificate. Generation and learned credit modules use policy-visible histories; snapshot costs count toward PWPO's training budget. Grouping baselines receive the same policy-visible histories, while their published grouping/discount choices remain method parameters. The reported method ranking uses this shared protocol. Published native-interface scores follow their respective benchmark protocols.

Additional feedback. BEACON (Wang et al., 2026b) uses intermediate milestone feedback. Its supplementary track supplies the identical detector and feedback to a GRPO control and to the trajectory-level LOO control, with equal tuning and resource budgets; the published temporal factor 0.95 and segment weight 1 are starting values. This track assesses learning with extra feedback. PWPO's theoretical reward-invariance condition applies to the terminal-reward track.

Mechanism controls. *Seq-LOO* assigns a singleton event to every rollout and shares PWPO's independent-rollout protocol, task/seed schedule, unnormalized leave-one-out advantages, optimizer, and full-trajectory clipped ratio. It implements LOO policy optimization at the trajectory level (Ahmadian et al., 2024; Schulman et al., 2017). *Uniform-Pair* equally averages the separate clipped losses of the two members in PWPO's certified cells; each original rollout retains total weight one. This augmentation heuristic can introduce estimator bias through equal weighting. A further control weights these separate losses by frozen old-policy conditional member probabilities, isolating memberwise clipping from PWPO's clipping after exact mass aggregation. All pair controls pay the same certification and rescoring costs. The end-task mechanism comparison uses AppWorld-Atomic at 7B with the 19,200-rollout budget and default $N = 8$, one update epoch, and zero KL coefficient (Table 8). G2PO supplies the strongest baseline in the main comparison; the controlled variants isolate pairing weights and the placement of clipping.

Finite-gradient validation. A finite, history-sensitive tool model provides an exact full-gradient reference by enumerating complete transcripts and independent LOO blocks with $N = 3$. The fixed task distribution includes successful and failed trajectories and nondegenerate rewards. Two certificate regimes use the same policy and task distribution: certified local pairs, and singleton cells obtained by disabling pair acceptance. Table 10 compares Seq-LOO, Uniform-Pair, Frozen-weight Pair, and PWPO at $\theta = \theta_{\text{old}}$. The enumeration tests the on-policy estimator identities and singleton fallback.

B.3 Metrics, budgets, and statistical analysis

Coverage and correctness. For n original rollouts, random-boundary coverage is

$$\widehat{p}_{\text{pair}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{|C_{j_i}(\tau_i)| = 2\}. \tag{B.1}$$

This denominator includes short, unsuccessful, and rejected trajectories. The audit records the original-rollout and accepted-pair counts, restore/original-replay faults, and independent recheck violations among accepted pairs. An accepted-pair violation means disagreement in associated call outputs, complete pair-end state, or final reward/termination status after replaying the full continuation. Coverage measures acceptance at the single randomly selected boundary of each rollout. All-task success remains primary even when coverage is negligible.

Gradient and cost diagnostics. The measured gradient/cost diagnostic is restricted to the AppWorld-Atomic/7B reference checkpoint underlying Table 7(b), at $\theta = \theta_{\text{old}}$ and $N = 8$. Table 11 reports uncertainty and the integer-budget comparison. Independent task blocks share the

original rollouts across estimators; each complete block is one statistical unit. Let $\mu_e = \mathbb{E}G_{e,N}$ and $\widetilde{V}_e = \mathbb{E}\|G_{e,N} - \mu_e\|^2$. For the unbiased estimators, $\widetilde{V}_e = V_e(N)$ from Section 4.4; Uniform-Pair's MSE also includes its squared bias. The diagnostic reports paired mean-gradient differences and ratios of centered variance, complete block cost, and their product, with paired block-bootstrap intervals. For a fixed diagnostic budget $\mathcal{B}_{\text{diag}}$, the prediction uses $K_e = \lfloor \mathcal{B}_{\text{diag}}/c_e \rfloor \geq 1$ and variance $\widetilde{V}_e/K_e$; it is an MSE prediction only for unbiased estimators. Cost calibration and block counts are fixed before evaluating the budgeted gradient averages. Full gradients are used on the finite model; the LLM diagnostic concerns the measured gradient representation.

Training budgets and cost accounting. The rollout-matched comparisons use 150 iterations of 16 tasks and 8 rollouts, or 19,200 original trajectories per run. The separate compute-matched comparison is restricted to AppWorld-Atomic/7B. A five-iteration training-only Seq-LOO pilot estimates per-iteration cost for this setting; 150 times that estimate defines the common node-time budget, frozen before comparative training. Pilot, development-search, evaluation, and final-training costs are accounted for separately. Within a comparison, identical hardware allocation is used and GPU-hours count allocated accelerators, including idle waiting for environment execution. Reported costs are allocated GPU-hours and total executed tool calls; the latter include rollout and certification/replay execution. Snapshots, restores, both replay orders, old/current likelihoods, backward passes, and baseline critics/grouping are charged. Table 7(a) compares final completion at the common compute budget, while Table 8 records the costs of the rollout-matched mechanism controls.

Uncertainty. The 1.5B runs use five seeds; 7B/14B runs use at least three. Tables summarize means and standard deviations across runs. Paired 95% intervals are restricted to the AppWorld-Atomic/7B mechanism comparison in Table 8, resampling matched training seeds and then scenarios while retaining all three variants. Each evaluation replicate uses one rollout per test task, with decoding seeds shared across methods. Replicates are averaged to estimate single-rollout success. Checkpoints are selected on development performance within the prescribed budget and then locked for test evaluation.

B.4 Backbones, optimization, and PWPO configuration

Models and hardware. The backbone checkpoints are `Qwen2.5-1.5B-Instruct`, `Qwen2.5-7B-Instruct`, and `Qwen2.5-14B-Instruct` (Qwen Team, 2024). The 1.5B/7B pair matches GiGPO, HGPO, BEACON, and G2PO's ALFWorld/WebShop comparisons; 14B matches G2PO's AppWorld scale. The main model matrix is 1.5B/7B on ALFWorld/WebShop, 7B/14B on AppWorld-Atomic, and 1.5B/7B on supplementary ScienceWorld. These checkpoints test scale transfer within the Qwen2.5 family. The configuration uses full-parameter fine-tuning, bfloat16, gradient checkpointing, and a veRL-style distributed trainer with vLLM generation on one node with eight NVIDIA H100 80GB GPUs.

Table 4: Default settings for the controlled track. Limits and budgets apply equally within each benchmark/backbone comparison. Method-specific baseline settings are recorded separately.

Setting	Default value
Tasks / independent rollouts	$B = 16$ / $N = 8$
Actor optimizer	AdamW; learning rate 10^{-6}
Optimizer parameters	$(\beta_1, \beta_2) = (0.9, 0.999)$; $\varepsilon_{\text{Adam}} = 10^{-8}$; weight decay 0
Clipping / gradient norm	$\epsilon = 0.2$ / 1.0
Update schedule	One epoch per rollout batch; minibatches of 64 original trajectories
PWPO / Seq-LOO advantage	Leave-one-out; no standard-deviation normalization; return discount 1
Auxiliary regularization	KL coefficient 0; entropy coefficient 0 in the controlled track
Generation temperature	Training 1.0; evaluation 0.4
Support and context	Fixed grammar; no top-k/top-p truncation; full history up to 32,768 total tokens
Response limit	512 generated tokens per invocation/submission
Boundary / cell	One uniform boundary; at most two distinct transcripts; no retry

Likelihood and update details. Generation uses the same normalized grammar-constrained distribution that is scored during training. All policy-generated text is scored, including terminal submissions and incomplete final prefixes. An episode that exhausts the fixed context/response limit receives its prescribed limit status, without sliding-window truncation or removal from the batch. For $m \geq 2$, exactly one boundary is uniform over all $m - 1$ possibilities; $m < 2$ gives the identity cell. The boundary index is retained, repeated samples preserve multiplicity, and accepted cells contain two distinct records. Stable log-sum-exp computes the exact mass; reordered-history suffixes are fully rescored. Cells, LOO advantages, and behavior-policy masses are frozen during the update; current masses are differentiated.

Configuration selection. Baseline-specific parameters start from the published or released configuration. Method-specific development searches use matched allocations and select configurations using development performance. PPO starts with critic learning rate 10^{-5}.

B.5 Preliminary study: protocol and diagnostics

The frozen-policy diagnostic in Section 3 measures variation between equivalent execution orders. The following definitions and aggregation rules underlie Figure 2.

B.5.1 Workflow pairs and likelihood evaluation

The dataset contains 640 equivalent order pairs: 176 independent read pairs, 165 independent query pairs, 168 cross-store lookups, and 131 disjoint writes. Each pair is one unit of analysis. The policy and rewards remain fixed throughout this diagnostic.

At an eligible boundary, u_0 and u_1 exchange two adjacent call–output blocks while preserving all other sampled tokens. Call independence follows from the controlled environment specification. The workflow semantics guarantee the same output for each call, the same pair-end state, termination status, and terminal reward; both orders are also replayed as a consistency check. Outputs are matched by call identity. Reordering changes the policy's visible history while preserving these environment outcomes.

For the same frozen policy and task x, write $\ell_r = \log p_\theta(u_r \mid x)$ and $\psi_r = \nabla_\theta \log p_\theta(u_r \mid x)$, for $r \in \{0, 1\}$. The complete likelihood includes every sampled token under its own history, including the unchanged suffix evaluated under the reordered context. Deterministic tool outputs enter the conditioning context with probability factor one (Appendix A.1.3). Each transcript has 64 recorded score coordinates. All norms, cosines, and second moments use these vectors before projection for visualization.

B.5.2 Order balance, score geometry, and conditional variation

The conditional order weights and order-balance statistic are

$$w_r = \frac{\exp(\ell_r)}{\exp(\ell_0) + \exp(\ell_1)}, \qquad B = 4w_0 w_1 \in [0, 1]. \tag{B.2}$$

Thus $B = 1$ means equally plausible orders, while $B \to 0$ means one order dominates. The weights describe the relative probabilities of the two orders conditional on their pair. The absolute log-probability gap is $|\Delta\ell| = |\ell_0 - \ell_1| = |\log w_0 - \log w_1|$, measured in nats.

Using $\varepsilon = 10^{-12}$, the normalized score divergence and score cosine are

$$\begin{aligned} d_\psi &= \frac{\|\psi_0 - \psi_1\|_2}{\frac{1}{2}(\|\psi_0\|_2 + \|\psi_1\|_2) + \varepsilon}, \\ c_\psi &= \frac{\psi_0^\top \psi_1}{\|\psi_0\|_2 \|\psi_1\|_2 + \varepsilon}. \end{aligned} \tag{B.3}$$

All recorded score norms are positive. Small d_ψ indicates similar vectors; the cosine measures their directional alignment up to the numerical stabilizer.

For each pair, let A be its recorded shared advantage, $g_r = A\psi_r$ its two gradient contributions, and $\bar{g} = w_0 g_0 + w_1 g_1$ their conditional mean. Define the order-dependent variation V and conditional

second moment M by

$$V = w_0\|g_0 - \bar{g}\|_2^2 + w_1\|g_1 - \bar{g}\|_2^2 = w_0 w_1\|g_0 - g_1\|_2^2,$$
$$M = w_0\|g_0\|_2^2 + w_1\|g_1\|_2^2 = V + \|\bar{g}\|_2^2. \tag{B.4}$$

The reported order-noise fraction is

$$\nu = \frac{V}{M + \varepsilon}. \tag{B.5}$$

All recorded pairs have $M > 0$. The numerator is the trace of the covariance induced by choosing an order within the pair, and the denominator is the pair's second moment. Since the same A multiplies both scores, its squared scale cancels apart from ε. The mean $\bar{g}$ is used only to compute the diagnostic. A pair's contribution to unconditional training variance also depends on its frequency and gradient magnitude.

B.5.3 PCA PROJECTION AND HEATMAP CONSTRUCTION

Paired gradient panel. Stack the 1,280 vectors $g_r = A\psi_r$ from both orders into a matrix with 64 columns, subtract each column's mean, and project onto the first two principal components. PCA assigns equal weight to every row, without feature standardization or per-vector normalization. The plotted coordinates agree with this projection up to the arbitrary sign of each component. The first two components explain 18.52% of total centered gradient variance. Segments connect each pair, with circles for the original order and triangles for the swapped order. Scalar diagnostics use all 64 coordinates.

Stratified noise panel. For B, the bin boundaries are 0, 0.2, 0.4, 0.6, 0.75, 0.9, and 1; for d_ψ, they are 0, 0.5, 0.7, 0.9, 1.1, 1.3, and 2. Every bin includes its lower endpoint and excludes its upper endpoint, except that the last bin includes the upper endpoint as well. Each cell shows the median ν and the number of pairs in that cell. All 640 records are included, and all 36 cells are occupied. Cells have equal display size, with the unequal numerical bin boundaries marked on the axes; counts give their frequencies. Cell percentages are rounded to whole percentages, while the text reports one decimal place. The shared color scale spans 0–60%, encoding individual-pair ν on segments and cell-median ν in the heatmap.

B.5.4 NUMERICAL CONSISTENCY

Consistency checks verify unique nonnegative integer pair identifiers, finite values, matching score dimensions, positive normalized weights, and valid metric ranges. Scalar diagnostics are recomputed from the paired vectors and checked against Equation (B.4); the heatmap counts sum to 640. These checks assess the internal consistency of the recorded score vectors and their summaries.

B.6 LAYERWISE ORDER-NOISE DIAGNOSTIC

This appendix details the diagnostic in Section 5.5 and Figure 5. We analyze 1,600 accepted two-member cells at the AppWorld-Atomic/Qwen2.5-7B-Instruct reference checkpoint used in Table 7(b), with $N = 8$ and $\theta = \theta_{\text{old}}$.

Removable mass. For a cell $c = (j, \{u_0, u_1\})$ with $m(c) \geq 2$ completed call–output blocks, its normalized boundary position is $p(c) = j/(m(c) - 1)$. Let $\psi^{(\ell)}(u_r)$ denote the full-transcript score restricted to depth bin ℓ, $A(c)$ the fixed LOO advantage, and $w_r(c)$ the within-cell conditional probability. The layerwise trace of the removable conditional covariance is

$$\Omega_\ell(c) = A(c)^2 w_0(c) w_1(c) \left\| \psi^{(\ell)}(u_0) - \psi^{(\ell)}(u_1) \right\|_2^2. \tag{B.6}$$

This is the depthwise decomposition of the two-member term in Proposition 1. All recorded cells have positive total mass over the $L = 28$ measured depth bins. We normalize within each cell and aggregate over workflow-position interval I_b:

$$\widetilde{\Omega}_\ell(c) = \frac{\Omega_\ell(c)}{\sum_{k=1}^{L} \Omega_k(c)}, \qquad M_{\ell,b} = \text{median}_{c:\, p(c) \in I_b} \widetilde{\Omega}_\ell(c). \tag{B.7}$$

Landscape and contours. The 12 position bins have edges $0, 0.08, \ldots, 0.88, 1$. Intervals are left-closed and right-open, with the last including 1. We smooth $100 M_{\ell,b}$ with a Gaussian filter of standard deviations 1 depth bin and 0.8 position bins, using nearest-edge extension. Each cell also contributes one location $(p(c), \ell^\star(c))$, where $\ell^\star(c) = \arg\max_\ell \widetilde{\Omega}_\ell(c)$, to a two-dimensional histogram. Histogram counts are divided by the number of cells and bin area, smoothed with the same filter, and renormalized to integrate to one. White contour heights are the 55th, 70th, 82nd, 92nd, and 97th percentiles of positive density-grid values.

C Additional Experimental Results

C.1 Longer-horizon WebShop

Table 5 increases the WebShop limit from 15 to 30 actions, following the longer horizon used by HGPO (He et al., 2026), while retaining the common atomic interface and 19,200-rollout budget. PWPO attains the highest mean score and success at both scales, exceeding G2PO's success by 2.24 points at 1.5B and 1.47 points at 7B.

Table 5: **WebShop at 30 actions.** Score and success (%, ↑) for all methods under the shared atomic interface and 19,200-rollout budget. The longer horizon tests sensitivity to the limit used by HGPO (He et al., 2026). Results are mean ± standard deviation; bold denotes the highest mean in each column.

	Qwen2.5-1.5B-Instruct		Qwen2.5-7B-Instruct	
Method	Score	Succ.	Score	Succ.
GiGPO	$78.02_{\pm 0.83}$	$63.16_{\pm 0.52}$	$89.42_{\pm 0.58}$	$77.67_{\pm 0.42}$
HGPO	$78.74_{\pm 0.71}$	$64.88_{\pm 0.48}$	$88.74_{\pm 0.52}$	$79.27_{\pm 0.58}$
G2PO	$81.12_{\pm 0.66}$	$67.00_{\pm 0.55}$	$90.86_{\pm 0.63}$	$80.53_{\pm 0.31}$
Seq-LOO	$77.48_{\pm 0.94}$	$62.00_{\pm 0.45}$	$88.31_{\pm 0.77}$	$76.53_{\pm 0.50}$
PWPO (ours)	$\mathbf{82.76_{\pm 0.62}}$	$\mathbf{69.24_{\pm 0.61}}$	$\mathbf{92.24_{\pm 0.48}}$	$\mathbf{82.00_{\pm 0.40}}$

C.2 ScienceWorld and feedback controls

PWPO achieves the highest mean score and completion rate among the terminal-reward methods in Table 6. Relative to G2PO, completion increases by 2.84/2.28 points at 1.5B/7B. BEACON (Wang et al., 2026b) attains higher absolute performance with additional milestone feedback; its comparisons use GRPO and Seq-LOO with the same detector. The feedback tracks are reported separately, and PWPO's reward-invariance guarantee applies to the terminal-reward track.

Table 6: **ScienceWorld results.** Native score and completion rate (%, ↑), with a 30-action limit and 19,200 training rollouts. Results are mean ± standard deviation. † denotes shared-milestone feedback; bold denotes the highest mean within each feedback track.

		Qwen2.5-1.5B-Instruct		Qwen2.5-7B-Instruct	
Method	Reward feedback	Score	Succ.	Score	Succ.
PPO	Terminal	$29.41_{\pm 1.62}$	$10.96_{\pm 1.10}$	$37.28_{\pm 1.35}$	$24.15_{\pm 1.22}$
RLOO	Terminal	$30.72_{\pm 1.48}$	$16.54_{\pm 1.31}$	$44.08_{\pm 1.27}$	$32.05_{\pm 1.11}$
GRPO	Terminal	$31.84_{\pm 1.36}$	$21.26_{\pm 1.18}$	$61.64_{\pm 1.06}$	$49.12_{\pm 1.02}$
GiGPO	Terminal	$35.71_{\pm 1.21}$	$25.86_{\pm 1.12}$	$69.31_{\pm 0.94}$	$53.55_{\pm 0.91}$
HGPO	Terminal	$39.06_{\pm 1.17}$	$28.74_{\pm 1.06}$	$72.18_{\pm 0.88}$	$55.43_{\pm 0.86}$
G2PO	Terminal	$43.27_{\pm 1.09}$	$32.64_{\pm 1.14}$	$75.46_{\pm 0.82}$	$57.36_{\pm 0.83}$
Seq-LOO	Terminal	$42.15_{\pm 1.12}$	$31.72_{\pm 1.08}$	$73.81_{\pm 0.90}$	$56.04_{\pm 0.88}$
GRPO†	Shared milestones	$54.23_{\pm 1.05}$	$41.06_{\pm 1.03}$	$79.42_{\pm 0.79}$	$60.58_{\pm 0.77}$
Seq-LOO†	Shared milestones	$56.71_{\pm 0.99}$	$43.28_{\pm 0.96}$	$81.18_{\pm 0.74}$	$62.11_{\pm 0.72}$
BEACON†	Shared milestones	$\mathbf{58.93_{\pm 0.91}}$	$\mathbf{45.34_{\pm 0.87}}$	$\mathbf{83.68_{\pm 0.68}}$	$\mathbf{64.27_{\pm 0.65}}$
PWPO (ours)	Terminal	$\mathbf{46.81_{\pm 0.97}}$	$\mathbf{35.48_{\pm 0.99}}$	$\mathbf{78.12_{\pm 0.76}}$	$\mathbf{59.64_{\pm 0.75}}$

C.3 COMPUTE-MATCHED PERFORMANCE AND ESTIMATOR DIAGNOSTICS

Table 7(a) evaluates AppWorld-Atomic at 7B under the common compute budget defined in Appendix B.3. PWPO obtains 26.59/22.30 TGC on Test-N/Test-C, compared with G2PO's 24.40/20.14, at 381.7 versus 383.1 mean GPU-hours. Its 213.9 thousand tool calls include certification replay, exceeding G2PO's 169.6 thousand; the gain at comparable GPU time comes with more tool interactions. All test tasks enter the success metrics, irrespective of certification coverage.

Table 7: **Performance under a common compute budget and estimator diagnostics.** AppWorld-Atomic, Qwen2.5-7B-Instruct. (a) Mean ± standard deviation of completion and resource use; bold marks the highest mean completion. (b) Task-block gradient diagnostics at $\theta = \theta_{\text{old}}$; bold marks the lowest variance and cost–variance ratios, including ties.

(a) Compute-matched task performance

	Test-N		Test-C		Measured cost	
Method	TGC	SGC	TGC	SGC	GPU-hours	Tool calls ($\times 10^3$)
PPO	$17.86_{\pm 1.19}$	$12.50_{\pm 1.79}$	$13.43_{\pm 0.48}$	$9.83_{\pm 0.42}$	$381.6_{\pm 2.7}$	$153.4_{\pm 3.1}$
RLOO	$19.84_{\pm 0.91}$	$14.29_{\pm 1.79}$	$15.19_{\pm 0.60}$	$11.27_{\pm 0.42}$	$382.4_{\pm 2.1}$	$183.7_{\pm 4.0}$
GRPO	$20.83_{\pm 0.60}$	$15.48_{\pm 1.03}$	$15.83_{\pm 0.48}$	$11.99_{\pm 0.42}$	$380.9_{\pm 2.4}$	$181.5_{\pm 3.6}$
GiGPO	$22.62_{\pm 0.60}$	$16.67_{\pm 1.03}$	$17.51_{\pm 0.48}$	$13.19_{\pm 0.42}$	$382.8_{\pm 2.0}$	$176.2_{\pm 3.4}$
HGPO	$23.21_{\pm 0.60}$	$17.26_{\pm 1.03}$	$18.23_{\pm 0.48}$	$13.67_{\pm 0.72}$	$381.9_{\pm 2.5}$	$172.8_{\pm 3.8}$
G2PO	$24.40_{\pm 0.60}$	$18.45_{\pm 1.03}$	$20.14_{\pm 0.24}$	$14.63_{\pm 0.42}$	$383.1_{\pm 1.8}$	$169.6_{\pm 3.1}$
Seq-LOO	$24.01_{\pm 0.91}$	$17.26_{\pm 1.03}$	$19.42_{\pm 0.72}$	$14.15_{\pm 0.42}$	$382.6_{\pm 2.2}$	$176.8_{\pm 3.5}$
PWPO (ours)	$\mathbf{26.59_{\pm 0.91}}$	$\mathbf{19.64_{\pm 1.79}}$	$\mathbf{22.30_{\pm 0.48}}$	$\mathbf{16.31_{\pm 0.42}}$	$381.7_{\pm 2.3}$	$213.9_{\pm 4.7}$

(b) Variance–cost decomposition

Estimator	$\widehat{p}_{\text{pair}}$ (%)	V/V_{Seq}	c/c_{Seq}	$cV/(c_{\text{Seq}}V_{\text{Seq}})$
Seq-LOO	0.0	1.00	1.00	1.00
Uniform-Pair	39.4	0.87	1.18	1.03
Frozen-weight Pair	39.4	**0.77**	1.18	**0.91**
PWPO (ours)	39.4	**0.77**	1.18	**0.91**

TGC/SGC are percentages (↑). GPU-hours count allocated accelerators; tool calls include rollout and certification replay. Coverage uses all original rollouts; Seq-LOO's zero denotes no pairing. V is task-block gradient variance and c is complete block cost. The last column is $(V/V_{\text{Seq}})(c/c_{\text{Seq}})$, rounded to two decimals. Ratios share the Seq-LOO reference and checkpoints.

Variance–cost trade-off. At 39.4% pair coverage, PWPO's variance ratio is 0.77 and its cost ratio is 1.18, yielding $0.77 \times 1.18 = 0.9086 \approx 0.91$ relative cost–variance. Uniform-Pair reduces variance to 0.87; its cost–variance ratio is $0.87 \times 1.18 = 1.0266 \approx 1.03$. Frozen-weight Pair and PWPO coincide at the reported precision. At this on-policy checkpoint, both probability-weighted estimators improve cost–variance relative to Seq-LOO.

Statistical interpretation. Reported gains are differences between seed means. Mean and standard deviation summarize the task-performance runs; panel (b) reports point estimates. The statistical unit, seed configuration, and checkpoint selection are specified in Appendix B.3.

C.4 END-TASK MECHANISM COMPARISON

The controls share the AppWorld-Atomic/7B initialization, data, 100-call limit, and 19,200 original rollouts. Uniform-Pair uses equal member weights. Frozen-weight Pair uses old-policy conditional weights and clips member ratios separately. PWPO clips the ratio of the aggregated cell mass. The measured costs include certification, replay, and full-transcript rescoring.

Table 8: **Mechanism ablation on AppWorld-Atomic/7B.** (a) TGC/SGC (%, ↑) and costs, reported as mean ± standard deviation; bold marks the highest mean completion. G2PO, Seq-LOO, and PWPO completion values match Table 1; all costs are measured at the same rollout budget. (b) PWPO-minus-comparator differences in percentage points, with paired 95% intervals in brackets.

(a) Task completion and complete training cost

	Test-N		Test-C		Measured cost	
Method	TGC	SGC	TGC	SGC	GPU-hours	Tool calls ($\times 10^3$)
G2PO	$24.80_{\pm 0.91}$	$18.45_{\pm 1.03}$	$20.38_{\pm 0.72}$	$14.87_{\pm 0.83}$	$397.9_{\pm 2.6}$	$176.1_{\pm 3.3}$
Seq-LOO	$24.01_{\pm 0.91}$	$17.26_{\pm 1.03}$	$19.42_{\pm 0.72}$	$14.15_{\pm 0.42}$	$382.6_{\pm 2.2}$	$176.8_{\pm 3.5}$
Uniform-Pair	$25.60_{\pm 1.03}$	$18.45_{\pm 1.03}$	$21.58_{\pm 0.48}$	$15.83_{\pm 0.72}$	$449.6_{\pm 3.1}$	$249.8_{\pm 5.0}$
Frozen-weight Pair	$26.39_{\pm 0.69}$	$20.24_{\pm 1.03}$	$22.30_{\pm 0.48}$	$16.55_{\pm 0.72}$	$450.8_{\pm 2.9}$	$251.1_{\pm 5.2}$
PWPO (ours)	$\mathbf{27.38_{\pm 1.19}}$	$\mathbf{20.83_{\pm 1.03}}$	$\mathbf{23.02_{\pm 0.72}}$	$\mathbf{17.03_{\pm 0.42}}$	$451.5_{\pm 3.0}$	$252.4_{\pm 5.3}$

(b) Paired differences: PWPO minus comparator

	Test-N		Test-C	
Comparison	ΔTGC	ΔSGC	ΔTGC	ΔSGC
PWPO − G2PO	2.58 [1.31, 3.84]	2.38 [0.71, 4.05]	2.64 [1.72, 3.57]	2.16 [0.94, 3.36]
PWPO − Seq-LOO	3.37 [2.05, 4.70]	3.57 [1.68, 5.39]	3.60 [2.54, 4.65]	2.88 [1.72, 4.05]
PWPO − Uniform-Pair	1.78 [0.55, 3.02]	2.38 [0.49, 4.28]	1.44 [0.49, 2.39]	1.20 [0.16, 2.23]
PWPO − Frozen-weight Pair	0.99 [−0.06, 2.04]	0.59 [−0.83, 2.02]	0.72 [−0.12, 1.55]	0.48 [−0.39, 1.35]

Resampling pairs methods by training seed and scenario, preserving the three task variants. Panel (a) uses rollout-matched costs; compute-matched costs appear in Table 7(a).

Table 8 places PWPO above G2PO, Seq-LOO, and Uniform-Pair on all four completion metrics, with paired intervals excluding zero. The gains over Frozen-weight Pair are 0.48–0.99 points, with all four intervals including zero. PWPO uses 451.5 GPU-hours at the fixed rollout budget, versus 382.6 for Seq-LOO; the compute-matched comparison is reported separately in Table 7(a).

C.5 CERTIFICATION AND REPLAY AUDIT

Table 9: **PWPO certification and replay audit.** Counts pool five training runs at 1.5B and three at 7B/14B, each with 19,200 original rollouts. The coverage denominator includes all original rollouts, including short, failed, and rejected records. Accepted cells undergo an independent replay recheck.

PWPO (ours)

Environment	Backbone	Original rollouts	Accepted pairs	Coverage (%)	Replay faults	Pair violations
AppWorld-Atomic	7B	57,600	22,694	39.4	0	0
AppWorld-Atomic	14B	57,600	23,962	41.6	0	0
ALFWorld	1.5B	96,000	28,896	30.1	0	0
ALFWorld	7B	57,600	18,259	31.7	0	0
WebShop	1.5B	96,000	22,944	23.9	0	0
WebShop	7B	57,600	14,285	24.8	0	0
ScienceWorld	1.5B	96,000	7,968	8.3	0	0
ScienceWorld	7B	57,600	5,702	9.9	0	0

Backbones are Qwen2.5-Instruct. Limits match Tables 1, 2, and 6 (WebShop: 15 actions). Replay faults count restore/original-replay failures and halt the affected update. Pair violations count accepted cells with mismatched associated outputs, complete pair-end state, or full-continuation reward/status. Accepted pairs count original records assigned to two-member cells, with multiplicity retained.

Table 9 reports coverage from 8.3% on ScienceWorld/1.5B to 41.6% on AppWorld-Atomic/14B. No restore/original-replay faults or accepted-pair violations were observed in the audited records.

C.6 FINITE-GRADIENT AND COST DIAGNOSTICS

For the finite model, $g = \nabla J$ is obtained by exact enumeration, $\mu_e = \mathbb{E}G_{e,3}$, $B_e = \|\mu_e - g\|^2$, and $\widetilde{V}_e = \mathbb{E}\|G_{e,3} - \mu_e\|^2$. The one-block MSE is $B_e + \widetilde{V}_e$. All ratios in Table 10 use the same nonzero Seq-LOO variance V_{Seq}.

Table 10: **Exact finite-gradient validation.** One fixed history-sensitive tool model, $N = 3$, and $\theta = \theta_{\mathrm{old}}$. Both certificate regimes use the same policy, task distribution, terminal rewards, and exact full-gradient reference. Bias, variance, and MSE are computed by exact enumeration of this finite model.

Estimator	Coverage (%)	B_e/V_{Seq}	$\widetilde{V}_e/V_{\mathrm{Seq}}$	$\mathrm{MSE}_e/V_{\mathrm{Seq}}$
Certified local pairs				
Seq-LOO	0.0	0.000	1.000	1.000
Uniform-Pair	42.6	0.034	0.842	0.876
Frozen-weight Pair	42.6	0.000	0.742	0.742
PWPO (ours)	42.6	0.000	0.742	0.742
Singleton fallback (pair acceptance disabled)				
Seq-LOO	0.0	0.000	1.000	1.000
Uniform-Pair	0.0	0.000	1.000	1.000
Frozen-weight Pair	0.0	0.000	1.000	1.000
PWPO (ours)	0.0	0.000	1.000	1.000

Each MSE ratio equals its bias-squared ratio plus its centered-variance ratio.

Table 11: **Uncertainty and finite-budget extension of Table 7(b).** AppWorld-Atomic/7B, $N = 8$, at the same recorded reference checkpoint. The variance, cost, and cost–variance point ratios match that table; brackets denote paired 95% block-bootstrap intervals.

Estimator	d_μ	$\widetilde{V}_e/V_{\mathrm{Seq}}$	c_e/c_{Seq}	Q_e/Q_{Seq}	K_e	$R_e^{\mathrm{var}}/R_{\mathrm{Seq}}^{\mathrm{var}}$
Seq-LOO	0	1.00	1.00	1.00	38	1.00
Uniform-Pair	0.0068 [0.0031, 0.0115]	0.87 [0.82, 0.93]	1.18 [1.15, 1.21]	1.03 [0.97, 1.10]	32	1.03 [0.97, 1.10]
Frozen-weight Pair	0.0044 [0.0020, 0.0077]	0.77 [0.71, 0.84]	1.18 [1.15, 1.21]	0.91 [0.83, 0.99]	32	0.91 [0.84, 1.00]
PWPO (ours)	0.0044 [0.0020, 0.0077]	0.77 [0.71, 0.84]	1.18 [1.15, 1.21]	0.91 [0.83, 0.99]	32	0.91 [0.84, 1.00]

$d_\mu = \|\overline{G}_e - \overline{G}_{\mathrm{Seq}}\|$ measures the finite-sample separation between paired mean gradients. $Q_e = c_e \widetilde{V}_e$, $K_e = \lfloor \mathcal{B}_{\mathrm{diag}}/c_e \rfloor \geq 1$, and $R_e^{\mathrm{var}} = \widetilde{V}_e/K_e$. Ratios use unrounded costs and variances; the integer-budget ratio is $(\widetilde{V}_e/V_{\mathrm{Seq}})(K_{\mathrm{Seq}}/K_e)$. For Uniform-Pair, MSE equals R_e^{var} plus squared bias. Reference-row zeros/ones are algebraic identities. Sampling variability contributes to d_μ.

Interpretation. In the finite model, Frozen-weight Pair and PWPO have zero squared bias at the reported precision and variance ratio 0.742; Uniform-Pair has squared-bias ratio 0.034. Disabling pair acceptance recovers the Seq-LOO statistics for every control. At the AppWorld checkpoint, Frozen-weight Pair and PWPO share a cost–variance ratio of 0.91 with interval $[0.83, 0.99]$. With 32 versus 38 blocks, their budgeted-variance ratio is also 0.91, with interval $[0.84, 1.00]$ reaching one at the reported precision.

C.7 Preliminary study: summary statistics

Each of the 640 pairs receives equal weight in dataset-level medians and quantiles; w_0, w_1 weight the two orders within each pair. The reported median ν is computed from the pairwise ratios. Quantiles use linear interpolation over sorted observations. The median and 90th percentile of $|\Delta\ell|$ are 1.66 and 3.15 nats. The median and 10th percentile of c_ψ are 0.63 and 0.28. The median and 90th percentile of ν are 8.9% and 23.3%.

The score-divergence quartiles, computed over all pairs, are $Q_{25} = 0.6760234804$ and $Q_{75} = 1.0386268509$ (rounded here). The high-balance/high-divergence group uses $B > 0.75$ and $d_\psi > Q_{75}$; its 48 pairs have median $\nu = 32.1\%$. The low-balance/low-divergence group uses $B < 0.35$ or $d_\psi < Q_{25}$; its 311 pairs have median $\nu = 4.2\%$. Selection uses the unrounded quartiles and strict inequalities. The groups are disjoint; the remaining 281 pairs enter the overall summaries and heatmap.

The heatmap describes how order variation is distributed across B and d_ψ. Both quantities and ν are computed from the same paired weights and score vectors.

C.8 Layerwise order-noise statistics

Table 12 reports the depth shares and peak locations. For a depth set D, the mean within-pair share is $100n^{-1}\sum_c \sum_{\ell \in D} \widetilde{\Omega}_\ell(c)$, with $n = 1{,}600$. For a position interval I, the total raw-mass share is $100 \sum_{c:\, p(c) \in I} \sum_\ell \Omega_\ell(c) \,/\, \sum_c \sum_\ell \Omega_\ell(c)$. The displayed intensity maximum and peak-density maximum occur in different position bins because the former summarizes layer shares and the latter counts dominant-bin locations.

Table 12: **Layerwise order-noise statistics.** Depth shares average within-pair proportions; the position share weights pairs by total removable mass. Peak locations use the smoothed arrays displayed in Figure 5.

Statistic	Value
Mean within-pair share, depth bins 15–28	73.91%
Mean within-pair share, depth bins 1–9	7.34%
Total raw mass at $p \in [0.32, 0.72)$	74.36%
Maximum displayed layer share	11.03%
Location of the displayed maximum	Bin 18, $p \in [0.72, 0.80)$
Location of maximum dominant-bin density	Bin 18, $p \in [0.56, 0.64)$

D PROOFS

D.1 PROJECTION AND ESTIMATOR GUARANTEES

D.1.1 LABELLED CELLS AND THE WEIGHTED PROJECTION

Fix x and suppress it in the notation. The augmented sample $z = (\tau, j)$ has probability $\widetilde{p}_\theta(z) = p_\theta(\tau)\nu(j \mid m(\tau))$. The map $\widetilde{F}(\tau, j) = (F_j\tau, j)$ is an involution, so its orbits $c = (j, C)$ partition the augmented space. Summing $\widetilde{p}_\theta$ over an orbit gives $q_\theta(c) = \nu(j \mid m(C))M_\theta(C)$. Dividing by this mass gives the member probability $w_\theta(u \mid c) = p_\theta(u)/M_\theta(C)$. Keeping j distinguishes orbits that share an unlabelled transcript.

Let the augmented space contain n samples and K cells. Define the membership matrix $H \in \{0,1\}^{n\times K}$ by $H_{zc} = 1$ exactly when $z \in c$, and let $D = \mathrm{diag}(\widetilde{p}_\theta)$. The conditional averaging operator is

$$\begin{aligned} \Pi &= H(H^\top DH)^{-1}H^\top D, \\ \Pi^2 = \Pi, \qquad \Pi^\top D &= D\Pi, \qquad \Pi\mathbf{1} = \mathbf{1}. \end{aligned} \tag{D.1}$$

The diagonal matrix $H^\top DH$ contains the positive cell probabilities. Consequently, Π projects onto $\mathrm{col}(H)$, the space of cell-constant functions, under the inner product $\langle f, g\rangle_D = f^\top Dg$. Applying it columnwise to the matrix of score vectors gives Equation (6). For a two-member cell with transcript probabilities p_0, p_1, its block has two identical rows $(p_0, p_1)/(p_0+p_1)$; the common boundary probability cancels. The projection acts on transcript-indexed functions, separately for each parameter coordinate.

D.1.2 PROOF OF THE MEAN AND COVARIANCE IDENTITY

For the fixed baseline in Proposition 1, let $G \in \mathbb{R}^{n\times d_\theta}$ have row $G_z = A(c)\psi_\theta(\tau)^\top$, where d_θ is the parameter dimension and $z = (\tau, j) \in c$. Reward and the baseline are constant on each cell, so $(\Pi G)_z = g_{\mathrm{pair}}^\top$. Weighted self-adjointness and $\Pi\mathbf{1} = \mathbf{1}$ imply $\mathbf{1}^\top D\Pi G = \mathbf{1}^\top DG$, preserving the mean. The expected score is zero by normalization; differentiating the finite reward expectation gives this common mean as $\nabla_\theta J_x(\theta)$.

Subtracting the second moments cancels the common mean outer product and yields

$$\begin{aligned} \mathrm{Cov}(g_{\mathrm{seq}}) - \mathrm{Cov}(g_{\mathrm{pair}}) &= G^\top D(I-\Pi)G \\ &= [(I-\Pi)G]^\top D[(I-\Pi)G] \succeq 0. \end{aligned} \tag{D.2}$$

The second equality follows from $\Pi^2 = \Pi$ and $\Pi^\top D = D\Pi$. For a two-member cell, the conditional covariance is $A(c)^2 w_0 w_1 \Delta\psi_C \Delta\psi_C^\top$; for a singleton it is zero. Averaging over cells gives Equation (7). In a parameter direction v, the reduction is strict exactly when $A(c)^2 w_0 w_1 (v^\top \Delta\psi_C)^2 > 0$ on cells with positive probability. Identical score vectors and zero advantages yield no reduction.

D.1.3 LEAVE-ONE-OUT BATCHES AND THE CLIPPED OBJECTIVE

For one task, let $z_1, \ldots, z_N$ be independent augmented rollouts from the behavior policy and let $c_i = (j_i, C_i)$ be their cells. Define $\widehat{g}_{\mathrm{seq}} = N^{-1}\sum_i A_i\psi_{\mathrm{old}}(\tau_i)$ and $\widehat{g}_{\mathrm{pair}} = N^{-1}\sum_i A_i\bar{\psi}_{\mathrm{old}}(c_i)$, using Equation (4). Every A_i is determined by $c_{1:N}$ because rewards are invariant within cells. Conditional on these cells, the member choices are independent. Hence

$$\begin{aligned} \widehat{g}_{\mathrm{pair}} &= \mathbb{E}[\widehat{g}_{\mathrm{seq}} \mid c_{1:N}], \\ \mathrm{Cov}(\widehat{g}_{\mathrm{seq}}) - \mathrm{Cov}(\widehat{g}_{\mathrm{pair}}) &= \mathbb{E}\left[\frac{1}{N^2}\sum_{i=1}^{N} A_i^2 w_{i0} w_{i1} \Delta\psi_i \Delta\psi_i^\top\right] \succeq 0. \end{aligned} \tag{D.3}$$

Here w_{ir} is the behavior-policy conditional weight of member r in cell i, and $\Delta\psi_i$ is its score-vector difference; singleton terms are zero. Each leave-one-out baseline is independent of its own rollout, so $\mathbb{E}[\widehat{g}_{\mathrm{seq}}] = \nabla J_x(\theta_{\mathrm{old}})$, and conditional averaging preserves this mean. The argument applies independently to each task in the batch and requires independently sampled complete rollouts, including their prefixes.

At any fixed candidate θ, the unclipped ratio is also exact away from the behavior policy. For a fixed baseline β, summing $q_{\mathrm{old}}(c)\rho_\theta(c)(R_x(c)-\beta)$ over cells gives $J_x(\theta)-\beta$. With a leave-one-out baseline, the same identity holds conditional on the other rollouts; the averaged baseline is constant in the current parameters. The covariance identities apply at $\theta=\theta_{\mathrm{old}}$, where every ratio equals one: clipping is locally inactive for $\epsilon>0$, and the gradient of Equation (5) is $\widehat{g}_{\mathrm{pair}}$.

D.2 Probability space and conditional score moments

This appendix proves the variance and budget results in Section 4.4, accounting for dependence among LOO summands and using prespecified block counts. The analysis uses the local cells defined by the certificate. We derive the U-statistic components and their covariances through the classical orthogonal decomposition of Hoeffding (1948).

Fix the policy at $\theta=\theta_{\mathrm{old}}$. The certificate, grammar, and support are parameter-independent; certification is exact and preserves the reward. We assume differentiation under the task expectation is valid, as it is for a finite task set.

A task block first samples x from the task distribution, then collects $N\geq 2$ independent augmented rollouts $z_i=(\tau_i,j_i)$. Write $e=\mathrm{s}$ for the sequence estimator and $e=\mathrm{p}$ for PWPO. For a single rollout, define

$$\begin{aligned} S_{\mathrm{s}}&=\psi_\theta(\tau), & S_{\mathrm{p}}&=\bar{\psi}_\theta(c),\\ Y&=R_x(\tau)-\mu_x, & \mu_x&=\mathbb{E}[R_x(\tau)\mid x],\\ \sigma_x^2&=\mathbb{E}[Y^2\mid x], & g_x&=\nabla J_x(\theta)=\mathbb{E}[YS_e\mid x]. \end{aligned} \tag{D.4}$$

Here $\mathbb{E}[S_e\mid x]=0$ for both estimators. A block returns $G_{e,N}=N^{-1}\sum_{i=1}^N A_iS_{e,i}$ with the LOO advantages in Equation (4), whose baselines are computed from the other sampled rollouts. Let $g=\mathbb{E}_x g_x=\nabla J(\theta)$ and introduce three nonnegative variance coefficients:

$$\begin{aligned} H&=\mathbb{E}_x\|g_x-g\|^2,\\ a_e&=\mathbb{E}_x\mathbb{E}[\|YS_e-g_x\|^2\mid x],\\ b_e&=\mathbb{E}_x\left[\sigma_x^2\mathbb{E}[\|S_e\|^2\mid x]+\|g_x\|^2\right]. \end{aligned} \tag{D.5}$$

All norms are Euclidean. H measures task heterogeneity; a_e and b_e describe within-task noise and the additional LOO interaction, respectively.

Required moments are $\mathbb{E}_x\|g_x\|^2$, $\mathbb{E}_x\mathbb{E}[Y^2\|S_e\|^2\mid x]$, and $\mathbb{E}_x[\sigma_x^2\mathbb{E}[\|S_e\|^2\mid x]]<\infty$ for $e\in\{\mathrm{s},\mathrm{p}\}$. These conditions hold, for example, under bounded rewards and uniformly bounded scores. All statements about matrices are in the positive-semidefinite order.

For later use define the conditional matrices

$$\begin{aligned} F_{e,x}&=\mathbb{E}[S_eS_e^\top\mid x], & M_{e,x}&=\mathbb{E}[Y^2S_eS_e^\top\mid x],\\ \Sigma_{e,x}&=M_{e,x}-g_xg_x^\top, & \mathsf{H}&=\mathrm{Cov}_x(g_x),\\ \Omega(c,x)&=\mathrm{Cov}(S_{\mathrm{s}}\mid c,x), & \omega(c,x)&=\mathrm{tr}\,\Omega(c,x). \end{aligned} \tag{D.6}$$

Thus $a_e=\mathbb{E}_x\,\mathrm{tr}\,\Sigma_{e,x}$, $b_e=\mathbb{E}_x\,\mathrm{tr}(\sigma_x^2F_{e,x}+g_xg_x^\top)$, and $H=\mathrm{tr}\,\mathsf{H}$. These population quantities characterize the estimator's risk; updates use the sampled LOO baselines.

Lemma 1 (Conditional projection and weighted second moments)**.** *For each task, $\mathbb{E}[S_e\mid x]=0$ and $\mathbb{E}[YS_e\mid x]=g_x$ for both e. For every nonnegative cell-measurable scalar $f(c,x)$ with finite relevant moments,*

$$\mathbb{E}[fS_{\mathrm{s}}S_{\mathrm{s}}^\top\mid x]-\mathbb{E}[fS_{\mathrm{p}}S_{\mathrm{p}}^\top\mid x]=\mathbb{E}[f\Omega(c,x)\mid x]\succeq 0. \tag{D.7}$$

On a two-member cell, $\Omega=w_0w_1\Delta\psi\Delta\psi^\top$, where $\Delta\psi=\psi_\theta(u_0)-\psi_\theta(u_1)$; on a singleton, $\Omega=0$.

Proof. The selector law sums to one for each transcript and has no parameter derivative. Consequently, normalization and reward differentiation give

$$\begin{aligned}\mathbb{E}[S_{\mathrm{s}} \mid x] &= \sum_{\tau,j} p_\theta(\tau \mid x)\nu(j \mid m(\tau))\nabla \log p_\theta(\tau \mid x) \\ &= \sum_{\tau} \nabla p_\theta(\tau \mid x) = \nabla \sum_{\tau} p_\theta(\tau \mid x) = 0, \\ \mathbb{E}[Y S_{\mathrm{s}} \mid x] &= \sum_{\tau} R_x(\tau)\nabla p_\theta(\tau \mid x) - \mu_x \mathbb{E}[S_{\mathrm{s}} \mid x] = \nabla J_x(\theta) = g_x.\end{aligned} \tag{D.8}$$

The common selector probability cancels within a cell, so $S_{\mathrm{p}} = \mathbb{E}[S_{\mathrm{s}} \mid c, x]$. Since Y is constant on the cell, the tower property proves both first-moment assertions for S_{p}. For any fixed vector v, conditional Cauchy–Schwarz gives the explicit contraction

$$\begin{aligned}\mathbb{E}[f(v^\top S_{\mathrm{p}})^2 \mid x] &= \mathbb{E}\big[f\{\mathbb{E}[v^\top S_{\mathrm{s}} \mid c, x]\}^2 \mid x\big] \\ &\le \mathbb{E}\big[f\mathbb{E}[(v^\top S_{\mathrm{s}})^2 \mid c, x]\mathbb{E}[1^2 \mid c, x] \mid x\big] \\ &= \mathbb{E}[f(v^\top S_{\mathrm{s}})^2 \mid x].\end{aligned} \tag{D.9}$$

For the exact difference, write $S_{\mathrm{s}} = S_{\mathrm{p}} + \varepsilon$ with $\mathbb{E}[\varepsilon \mid c, x] = 0$. Conditional expansion eliminates the two cross terms and leaves $\mathbb{E}[\varepsilon\varepsilon^\top \mid c, x] = \Omega$, proving Equation (D.7). In a two-member cell, $S_{\mathrm{s}} - S_{\mathrm{p}}$ is $w_1\Delta\psi$ with probability w_0 and $-w_0\Delta\psi$ with probability w_1. Hence

$$\Omega = w_0 w_1^2 \Delta\psi\Delta\psi^\top + w_1 w_0^2 \Delta\psi\Delta\psi^\top = w_0 w_1(w_0 + w_1)\Delta\psi\Delta\psi^\top = w_0 w_1 \Delta\psi\Delta\psi^\top. \tag{D.10}$$

A singleton has zero conditional residual. This completes the proof. □

D.3 Exact LOO decomposition and variance reduction

Theorem 2 (Exact LOO risk and removable order noise)**.** *Average $K \ge 1$ independent task blocks of width N, obtaining $\widehat{g}_{e,K,N}$. Under the conditions above, both estimators are unbiased and their mean squared errors are exactly*

$$\mathbb{E}\|\widehat{g}_{e,K,N} - g\|^2 = \frac{V_e(N)}{K}, \qquad V_e(N) = H + \frac{a_e}{N} + \frac{b_e}{N(N-1)}. \tag{D.11}$$

For a two-member cell, set $\omega(c, x) = w_0 w_1 \|\psi_\theta(u_0) - \psi_\theta(u_1)\|^2$, and set $\omega = 0$ on singletons. Then

$$\begin{aligned}D_a &:= a_{\mathrm{s}} - a_{\mathrm{p}} = \mathbb{E}_{x,c}[Y(c)^2\omega(c, x)] \ge 0, \\ D_b &:= b_{\mathrm{s}} - b_{\mathrm{p}} = \mathbb{E}_{x,c}[\sigma_x^2\omega(c, x)] \ge 0, \\ \Delta_N &:= V_{\mathrm{s}}(N) - V_{\mathrm{p}}(N) = \frac{D_a}{N} + \frac{D_b}{N(N-1)}.\end{aligned} \tag{D.12}$$

The improvement is strict precisely when $D_a + D_b > 0$. The task-heterogeneity term H is unchanged.

We first fix x and suppress it. Let (Y_i, S_i) be independent copies of either (Y, S_{s}) or (Y, S_{p}), with $\mathbb{E}Y = 0$, $\mathbb{E}S = 0$, and $\mathbb{E}[YS] = g_x$. The use of a common task is essential: independent blocks sample their tasks anew, whereas all rollouts within one block condition on the same task.

Lemma 2 (LOO as an orthogonally decomposed U-statistic)**.** *The LOO block estimator has the following exact representation:*

$$\begin{aligned}G_N &= \binom{N}{2}^{-1} \sum_{i<k} h(z_i, z_k), & h(z, z') &= \tfrac{1}{2}(Y - Y')(S - S'), \\ G_N - g_x &= \frac{2}{N}\sum_{i=1}^{N} h_1(z_i) + \binom{N}{2}^{-1} \sum_{i<k} h_2(z_i, z_k), & & \\ h_1(z) &= \tfrac{1}{2}(YS - g_x), & h_2(z, z') &= -\tfrac{1}{2}(YS' + Y'S).\end{aligned} \tag{D.13}$$

Every first-order term is orthogonal in L^2 to every second-order term. Distinct second-order terms are mutually orthogonal, including pairs sharing one observation.

Proof. LOO advantages are invariant to subtracting μ_x. Expanding the ordered double sum and pairing its two orientations gives

$$\begin{aligned} G_N &= \frac{1}{N}\sum_i Y_i S_i - \frac{1}{N(N-1)}\sum_i\sum_{k\neq i} Y_k S_i \\ &= \frac{1}{N(N-1)}\sum_{i<k}(Y_i S_i + Y_k S_k - Y_k S_i - Y_i S_k) \\ &= \frac{1}{N(N-1)}\sum_{i<k}(Y_i - Y_k)(S_i - S_k) \\ &= \binom{N}{2}^{-1}\sum_{i<k} h(z_i, z_k). \end{aligned} \tag{D.14}$$

Independence and the zero means imply $\mathbb{E}h = g_x$ and

$$\begin{aligned} \mathbb{E}[h(z,z') \mid z] - g_x &= \tfrac{1}{2}(YS + \mathbb{E}[Y'S'] - Y\mathbb{E}S' - S\mathbb{E}Y') - g_x \\ &= \tfrac{1}{2}(YS - g_x) = h_1(z), \\ h(z,z') - g_x - h_1(z) - h_1(z') &= -\tfrac{1}{2}(YS' + Y'S) = h_2(z,z'), \\ \mathbb{E}[h_2(z,z') \mid z] &= -\tfrac{1}{2}(Y\mathbb{E}S' + S\mathbb{E}Y') = 0. \end{aligned} \tag{D.15}$$

Summing the decomposition of h counts each $h_1(z_i)$ exactly $N-1$ times and proves Equation (D.13).

We establish orthogonality by considering index overlaps. Disjoint index sets are independent and centered. If a first-order term and a second-order term share z_i, conditional degeneracy gives

$$\mathbb{E}[h_1(z_i)h_2(z_i,z_k)^\top] = \mathbb{E}\big[h_1(z_i)\mathbb{E}[h_2(z_i,z_k) \mid z_i]^\top\big] = 0. \tag{D.16}$$

If two different second-order pairs share z_i, their remaining observations are conditionally independent given z_i. Thus, for distinct i,k,l,

$$\mathbb{E}[h_2(z_i,z_k)h_2(z_i,z_l)^\top] = \mathbb{E}\big[\mathbb{E}[h_2(z_i,z_k) \mid z_i]\mathbb{E}[h_2(z_i,z_l) \mid z_i]^\top\big] = 0. \tag{D.17}$$

The reversed products vanish by transposition. These cases exhaust all possible overlaps and establish matrix-valued orthogonality. □

Lemma 3 (Exact conditional block covariance)**.** *For* $e \in \{\mathrm{s},\mathrm{p}\}$,

$$\mathbb{E}[G_{e,N} \mid x] = g_x, \qquad \mathrm{Cov}(G_{e,N} \mid x) = \frac{\Sigma_{e,x}}{N} + \frac{\sigma_x^2 F_{e,x} + g_x g_x^\top}{N(N-1)}. \tag{D.18}$$

Proof. The first identity follows from $\mathbb{E}h = g_x$. The first-order component has covariance $\Sigma_{e,x}/4$. To compute the second-order component, expand every product, using independence between primed and unprimed observations:

$$\begin{aligned} 4\mathbb{E}[h_2 h_2^\top \mid x] &= \mathbb{E}[Y^2 S'S'^\top + Y'^2 SS^\top + YY'S'S^\top + YY'SS'^\top \mid x] \\ &= \mathbb{E}[Y^2 \mid x]\mathbb{E}[S'S'^\top \mid x] + \mathbb{E}[Y'^2 \mid x]\mathbb{E}[SS^\top \mid x] \\ &\quad + \mathbb{E}[Y'S' \mid x]\mathbb{E}[YS \mid x]^\top + \mathbb{E}[YS \mid x]\mathbb{E}[Y'S' \mid x]^\top \\ &= 2\sigma_x^2 F_{e,x} + 2g_x g_x^\top. \end{aligned} \tag{D.19}$$

By Lemma 2, all unequal-index cross covariances vanish. Keeping their exact multiplicities gives

$$\begin{aligned} \mathrm{Cov}(G_{e,N} \mid x) &= \frac{4}{N^2}\sum_{i=1}^N \mathrm{Cov}(h_1(z_i) \mid x) + \binom{N}{2}^{-2}\sum_{i<k}\mathrm{Cov}(h_2(z_i,z_k) \mid x) \\ &= \frac{4}{N}\frac{\Sigma_{e,x}}{4} + \frac{2}{N(N-1)}\frac{\sigma_x^2 F_{e,x} + g_x g_x^\top}{2} \\ &= \frac{\Sigma_{e,x}}{N} + \frac{\sigma_x^2 F_{e,x} + g_x g_x^\top}{N(N-1)}. \end{aligned} \tag{D.20}$$

The term with denominator $N(N-1)$ gives the exact contribution of shared-baseline dependence. □

Complete proof of Theorem 2. First apply total covariance to a block, then independence to its K replicates. With $G_{e,N}^{(k)}$ denoting the kth independent block, the full matrix identity is

$$\begin{aligned}\mathrm{Cov}(\widehat{g}_{e,K,N}) &= \frac{1}{K^2}\sum_{k=1}^{K}\mathrm{Cov}(G_{e,N}^{(k)}) \\ &= \frac{1}{K}\left\{\mathrm{Cov}_x(\mathbb{E}[G_{e,N}\mid x]) + \mathbb{E}_x\,\mathrm{Cov}(G_{e,N}\mid x)\right\} \\ &= \frac{1}{K}\left\{\mathsf{H} + \frac{\mathbb{E}_x\Sigma_{e,x}}{N} + \frac{\mathbb{E}_x[\sigma_x^2 F_{e,x} + g_x g_x^\top]}{N(N-1)}\right\}.\end{aligned} \tag{D.21}$$

Its mean is g, so taking the trace proves Equation (D.11). This identity also proves nonnegativity of every variance coefficient from its defining covariance or second moment.

Apply Lemma 1 with $f = Y^2$ and $f = 1$. Because both estimators have the same g_x, their common mean outer product cancels, yielding

$$\begin{aligned}\Sigma_{\mathrm{s},x} - \Sigma_{\mathrm{p},x} &= \mathbb{E}[Y^2\Omega(c,x)\mid x] \succeq 0, \\ F_{\mathrm{s},x} - F_{\mathrm{p},x} &= \mathbb{E}[\Omega(c,x)\mid x] \succeq 0, \\ \mathrm{Cov}(G_{\mathrm{s},N}\mid x) - \mathrm{Cov}(G_{\mathrm{p},N}\mid x) &= \frac{1}{N}\mathbb{E}\left[\left(Y^2 + \frac{\sigma_x^2}{N-1}\right)\Omega(c,x)\mid x\right] \succeq 0.\end{aligned} \tag{D.22}$$

Integrate over x and take traces to obtain Equation (D.12). Since both coefficients of D_a, D_b are strictly positive for every finite $N \geq 2$, strict trace improvement is equivalent to $D_a + D_b > 0$.

An independent conditional-variance calculation verifies that the result matches the actual LOO baseline. Conditional on all cells, member choices are independent and each A_i is fixed. Consequently,

$$\begin{aligned}\mathbb{E}[G_{\mathrm{s},N}\mid c_{1:N}, x] &= G_{\mathrm{p},N}, \\ \mathrm{Cov}(G_{\mathrm{s},N}\mid c_{1:N}, x) &= \frac{1}{N^2}\sum_i A_i^2\Omega(c_i,x), \\ \mathbb{E}[A_i^2\mid c_i, x] &= \mathbb{E}\left[\left(Y_i - \frac{1}{N-1}\sum_{k\neq i}Y_k\right)^2 \,\middle|\, c_i, x\right] \\ &= Y_i^2 + \frac{1}{(N-1)^2}\sum_{k\neq i}\mathbb{E}[Y_k^2\mid x] = Y_i^2 + \frac{\sigma_x^2}{N-1}.\end{aligned} \tag{D.23}$$

The cross terms in the last expansion vanish by conditional independence and centering. Averaging the preceding conditional covariance recovers Equation (D.22) exactly.

For completeness, in every parameter direction v, the gain obeys the following chain, with an explicit equality criterion:

$$\begin{aligned}0 &\leq \frac{1}{N}\mathbb{E}_{x,c}\left[\left(Y^2 + \frac{\sigma_x^2}{N-1}\right)w_0 w_1 (v^\top\Delta\psi)^2\right] \\ &\leq \frac{\|v\|^2}{N}\mathbb{E}_{x,c}\left[\left(Y^2 + \frac{\sigma_x^2}{N-1}\right)w_0 w_1\|\Delta\psi\|^2\right] = \|v\|^2\Delta_N.\end{aligned} \tag{D.24}$$

The first expression is zero exactly when its nonnegative integrand is zero almost surely. Conditioning preserves g_x and therefore H. □

D.4 Finite budgets and exact sample counts

Let $c_e(N) > 0$ be the expected total cost of one complete task block, in a fixed additive resource unit. For PWPO, this includes sampling, snapshots, replay, both behavior and current likelihood evaluations needed for the estimator, and differentiation. Ordinary rejected certificates incur their

actual cost. The cost may depend on the same trajectories as the gradient. Given an expected budget $\mathcal{B}$, choose the block count *before collecting data*:

$$K_e(\mathcal{B}, N) = \left\lfloor \frac{\mathcal{B}}{c_e(N)} \right\rfloor, \qquad \mathcal{R}_e(\mathcal{B}, N) = \frac{V_e(N)}{K_e(\mathcal{B}, N)}. \tag{D.25}$$

Only affordable widths, with $K_e \geq 1$, are considered. The block count is fixed, and executing all blocks has expected cost at most $\mathcal{B}$.

Theorem 3 (Exact budget comparison and break-even cost)**.** *Under Theorem 2, $\mathcal{R}_e$ in Equation* (D.25) *is the exact MSE at the stated expected budget. At a common width with $V_{\mathrm{s}}(N) > 0$, PWPO is strictly more accurate if and only if*

$$\frac{\Delta_N}{V_{\mathrm{s}}(N)} > 1 - \frac{K_{\mathrm{p}}(\mathcal{B}, N)}{K_{\mathrm{s}}(\mathcal{B}, N)}. \tag{D.26}$$

For every $\mathcal{B} > c_e(N)$,

$$\frac{c_e(N)V_e(N)}{\mathcal{B}} \leq \mathcal{R}_e(\mathcal{B}, N) \leq \frac{c_e(N)V_e(N)}{\mathcal{B} - c_e(N)}. \tag{D.27}$$

Consequently, the leading budget coefficient is $Q_e(N) = c_e(N)V_e(N)$. If $c_{\mathrm{p}}(N) = (1+\kappa_N)c_{\mathrm{s}}(N)$ with $\kappa_N \geq 0$, then

$$Q_{\mathrm{p}}(N) < Q_{\mathrm{s}}(N) \quad \Longleftrightarrow \quad \frac{\Delta_N}{V_{\mathrm{s}}(N)} > \frac{\kappa_N}{1+\kappa_N}. \tag{D.28}$$

For any fixed nonempty admissible width family $\mathcal{N} \subseteq \{2, 3, \ldots\}$ with finite $c_e(N)$ and $V_e(N)$, define $Q_e^\star = \inf_{N \in \mathcal{N}} Q_e(N)$. Allowing each estimator to choose its own affordable width gives

$$\lim_{\mathcal{B}\to\infty} \mathcal{B} \inf_{\substack{N \in \mathcal{N} \\ c_e(N) \leq \mathcal{B}}} \mathcal{R}_e(\mathcal{B}, N) = Q_e^\star. \tag{D.29}$$

Each estimator optimizes its own width, so the comparison uses $Q_{\mathrm{s}}^\star$ for the sequence estimator.

Let $C_{e,N}^{(k)}$ be the realized nonnegative cost of block k, with $\mathbb{E}C_{e,N}^{(k)} = c_e(N)$. Costs may be correlated with block gradients; complete blocks are independent replicas. Widths and block counts are fixed before collecting the estimation data. An independent pilot may select a width, in which case the risk identity applies conditional on that choice. The budget guarantee requires the true mean cost or a valid deterministic upper bound on it. Tuning data are kept independent of the estimation data. The optimized comparisons are population benchmarks; estimating their coefficients requires a separate cost allowance.

Complete proof of Theorem 3. Exact risk under a planned expected budget. For the deterministic count $K = K_e(\mathcal{B}, N)$, linearity of expectation, the floor definition, and Theorem 2 give

$$\begin{aligned} \mathbb{E}\left[\sum_{k=1}^{K} C_{e,N}^{(k)}\right] &= \sum_{k=1}^{K} \mathbb{E}C_{e,N}^{(k)} = Kc_e(N) \\ &= c_e(N)\left\lfloor \frac{\mathcal{B}}{c_e(N)} \right\rfloor \leq c_e(N)\frac{\mathcal{B}}{c_e(N)} = \mathcal{B}, \\ \mathbb{E}\|\widehat{g}_{e,K,N} - g\|^2 &= \frac{V_e(N)}{K} = \mathcal{R}_e(\mathcal{B}, N). \end{aligned} \tag{D.30}$$

Linearity of expectation allows correlation between each block's cost and gradient.

Necessary and sufficient finite-budget comparison. Write $V_{\mathrm{s}} = V_{\mathrm{s}}(N) > 0$ and $K_e = K_e(\mathcal{B}, N) \geq 1$. Every denominator below is positive. Substitution followed by reversible algebra yields

$$\begin{aligned} \mathcal{R}_{\mathrm{p}} < \mathcal{R}_{\mathrm{s}} &\Longleftrightarrow \frac{V_{\mathrm{s}} - \Delta_N}{K_{\mathrm{p}}} < \frac{V_{\mathrm{s}}}{K_{\mathrm{s}}} \\ &\Longleftrightarrow K_{\mathrm{s}}V_{\mathrm{s}} - K_{\mathrm{s}}\Delta_N < K_{\mathrm{p}}V_{\mathrm{s}} \\ &\Longleftrightarrow K_{\mathrm{s}}\Delta_N > (K_{\mathrm{s}} - K_{\mathrm{p}})V_{\mathrm{s}} \\ &\Longleftrightarrow \frac{\Delta_N}{V_{\mathrm{s}}} > 1 - \frac{K_{\mathrm{p}}}{K_{\mathrm{s}}}. \end{aligned} \tag{D.31}$$

If $V_{\mathrm{s}} = 0$, Theorem 2 implies $V_{\mathrm{p}} = 0$, so the risks are equal.

Tight leading coefficient and the floor correction. For $t = \mathcal{B}/c_e(N) > 1$, $0 < t - 1 < \lfloor t \rfloor \leq t$. Taking reciprocals reverses the positive inequalities. Multiplication by $V_e(N) \geq 0$ gives the entire chain

$$\frac{c_e(N)V_e(N)}{\mathcal{B}} = \frac{V_e(N)}{t} \leq \frac{V_e(N)}{\lfloor t \rfloor} \leq \frac{V_e(N)}{t-1} = \frac{c_e(N)V_e(N)}{\mathcal{B} - c_e(N)}. \tag{D.32}$$

The lower bound is attained whenever t is an integer. The upper denominator is a universal floor relaxation; the exact expression in Equation (D.25) is sharper whenever its integer count is known. Multiplying Equation (D.32) by $\mathcal{B}$ and taking the limit proves $\mathcal{B}\mathcal{R}_e \to Q_e(N)$. When $c_{\mathrm{p}} = (1 + \kappa_N)c_{\mathrm{s}}$, another chain of equivalent inequalities proves Equation (D.28):

$$\begin{aligned} Q_{\mathrm{p}} < Q_{\mathrm{s}} &\iff (1+\kappa_N)(V_{\mathrm{s}} - \Delta_N) < V_{\mathrm{s}} \\ &\iff (1+\kappa_N)\Delta_N > \kappa_N V_{\mathrm{s}} \\ &\iff \Delta_N / V_{\mathrm{s}} > \kappa_N/(1+\kappa_N). \end{aligned} \tag{D.33}$$

Strictly unequal leading coefficients imply the corresponding risk ordering for all sufficiently large budgets at these fixed widths. With equal coefficients, finite-budget ordering depends on the floor terms.

Optimization over the width. Let $\mathcal{R}_e^\star(\mathcal{B})$ be the infimum in Equation (D.29). For every affordable N, the left side of Equation (D.32) remains valid even at $\mathcal{B} = c_e(N)$. Thus

$$\mathcal{B}\mathcal{R}_e^\star(\mathcal{B}) \geq \inf_{\substack{N \in \mathcal{N} \\ c_e(N) \leq \mathcal{B}}} Q_e(N) \geq \inf_{N \in \mathcal{N}} Q_e(N) = Q_e^\star \geq 0. \tag{D.34}$$

Assume $\mathcal{N}$ is nonempty. For any $\varepsilon > 0$, choose a fixed $N_\varepsilon \in \mathcal{N}$ with $Q_e(N_\varepsilon) \leq Q_e^\star + \varepsilon$. Since its cost is finite, it is affordable for all sufficiently large budgets. For such budgets,

$$\begin{aligned} Q_e^\star &\leq \liminf_{\mathcal{B}\to\infty} \mathcal{B}\mathcal{R}_e^\star(\mathcal{B}) \\ &\leq \limsup_{\mathcal{B}\to\infty} \mathcal{B}\mathcal{R}_e^\star(\mathcal{B}) \\ &\leq \lim_{\mathcal{B}\to\infty} \mathcal{B}\mathcal{R}_e(\mathcal{B}, N_\varepsilon) = Q_e(N_\varepsilon) \leq Q_e^\star + \varepsilon. \end{aligned} \tag{D.35}$$

Letting $\varepsilon \downarrow 0$ proves the limit, including when the infimum $Q_e^\star$ is unattained at finite widths. □

Complete proof of Theorem 1. Theorem 2 establishes that $G_{e,N}$ is unbiased, identifies its variance with $V_e(N)$ in Equation (D.11), and gives $V_{\mathrm{p}}(N) \leq V_{\mathrm{s}}(N)$ by Equation (D.12). For the deterministic positive count K_e, Equation (D.30) proves both the expected-budget guarantee and $\mathcal{R}_e = V_e/K_e$. Since $K_{\mathrm{s}}, K_{\mathrm{p}}$ and V_{s} are positive, the reversible inequalities in Equation (D.31) establish the equivalence in Equation (8). For fixed N, Equation (D.32) gives $\mathcal{B}\mathcal{R}_e \to c_e V_e = Q_e$. Substituting $c_{\mathrm{p}} = (1+\kappa_N)c_{\mathrm{s}}$ in Equation (D.33) proves Equation (9). If $Q_{\mathrm{p}} < Q_{\mathrm{s}}$, convergence to these distinct limits implies $\mathcal{R}_{\mathrm{p}} < \mathcal{R}_{\mathrm{s}}$ for all sufficiently large budgets. □

Corollary 1 (Exact accuracy-dependent sample and cost requirements)**.** *Fix a width and an RMS error tolerance $\zeta > 0$. Among unweighted averages of a deterministic positive number of independent blocks, the smallest block count, original rollout count, and expected cost attaining MSE at most ζ^2 are respectively*

$$K_e^{\min}(\zeta, N) = \max\{1, \lceil V_e(N)/\zeta^2 \rceil\}, \qquad n_e^{\min} = N K_e^{\min}, \qquad \mathcal{C}_e^{\min} = c_e(N) K_e^{\min}. \tag{D.36}$$

At a common width, $K_{\mathrm{p}}^{\min} \leq K_{\mathrm{s}}^{\min}$ and $n_{\mathrm{p}}^{\min} \leq n_{\mathrm{s}}^{\min}$. If $V_e(N) > 0$, then $\lim_{\zeta \downarrow 0} \zeta^2 \mathcal{C}_e^{\min} = Q_e(N)$.

Proof. For any positive integer K, exact risk gives

$$\mathbb{E}\|\widehat{g}_{e,K,N} - g\|^2 \leq \zeta^2 \iff V_e(N)/K \leq \zeta^2 \iff K \geq V_e(N)/\zeta^2. \tag{D.37}$$

Taking the smallest admissible integer proves Equation (D.36). Monotonicity of the ceiling and $V_{\mathrm{p}} \leq V_{\mathrm{s}}$ prove the sample-count statements. For $V_e(N) > 0$, the ceiling bounds yield

$$Q_e(N) \leq \zeta^2 c_e(N) \left\lceil \frac{V_e(N)}{\zeta^2} \right\rceil < Q_e(N) + \zeta^2 c_e(N). \tag{D.38}$$

The squeeze theorem proves the limit. When $V_e = 0$, one block already has zero error. Integer rounding can give the same sample count despite lower variance. Total cost also includes replay and scoring work. □

Proposition 2 (Comparison with the best sequence width)**.** *At a fixed budget, let $r_{\mathrm{s}}^{\star} > 0$ denote the infimum of the sequence risk over its affordable widths in $\mathcal{N}$. For a PWPO width with $V_{\mathrm{p}}(N) > 0$, define $k_{\star} = \lfloor V_{\mathrm{p}}(N)/r_{\mathrm{s}}^{\star} \rfloor + 1$. Then*

$$\mathcal{R}_{\mathrm{p}}(\mathcal{B}, N) < r_{\mathrm{s}}^{\star} \quad \Longleftrightarrow \quad c_{\mathrm{p}}(N) \le \mathcal{B}/k_{\star}. \tag{D.39}$$

For additive overhead $o_N = c_{\mathrm{p}}(N) - c_{\mathrm{s}}(N)$, the exact finite-budget threshold is $o_N \le \mathcal{B}/k_{\star} - c_{\mathrm{s}}(N)$. The strict leading-coefficient comparison is

$$Q_{\mathrm{p}}(N) < Q_{\mathrm{s}}^{\star} \quad \Longleftrightarrow \quad o_N < \frac{Q_{\mathrm{s}}^{\star}}{V_{\mathrm{p}}(N)} - c_{\mathrm{s}}(N). \tag{D.40}$$

If $H > 0$, the necessary same-width leading-order condition is $c_{\mathrm{p}}(N)H < c_{\mathrm{s}}(N)V_{\mathrm{s}}(N)$.

Proof. The count is integer, so strict risk improvement is equivalent to an integer threshold:

$$\begin{aligned} \frac{V_{\mathrm{p}}(N)}{\lfloor \mathcal{B}/c_{\mathrm{p}}(N) \rfloor} < r_{\mathrm{s}}^{\star} &\Longleftrightarrow \left\lfloor \frac{\mathcal{B}}{c_{\mathrm{p}}(N)} \right\rfloor > \frac{V_{\mathrm{p}}(N)}{r_{\mathrm{s}}^{\star}} \\ &\Longleftrightarrow \left\lfloor \frac{\mathcal{B}}{c_{\mathrm{p}}(N)} \right\rfloor \ge k_{\star} \\ &\Longleftrightarrow \frac{\mathcal{B}}{c_{\mathrm{p}}(N)} \ge k_{\star} \Longleftrightarrow c_{\mathrm{p}}(N) \le \frac{\mathcal{B}}{k_{\star}}. \end{aligned} \tag{D.41}$$

The right side also ensures affordability. Subtracting $c_{\mathrm{s}}(N)$ proves the overhead threshold; the weak inequality follows from the floor. Equation (D.40) follows by dividing the strict coefficient inequality by $V_{\mathrm{p}}(N) > 0$. If $r_{\mathrm{s}}^{\star} = 0$, strict improvement over it is impossible. If $V_{\mathrm{p}}(N) = 0 < r_{\mathrm{s}}^{\star}$, any affordable PWPO block suffices.

Finally, nonnegative within-task terms yield

$$0 \le \Delta_N = V_{\mathrm{s}}(N) - V_{\mathrm{p}}(N) \le V_{\mathrm{s}}(N) - H, \qquad Q_{\mathrm{p}}(N) \ge c_{\mathrm{p}}(N)H. \tag{D.42}$$

Strict improvement $Q_{\mathrm{p}}(N) < Q_{\mathrm{s}}(N)$ therefore requires $c_{\mathrm{p}}(N)H < Q_{\mathrm{s}}(N)$. The finite-budget analogue follows from $\mathcal{R}_{\mathrm{p}} \ge H/K_{\mathrm{p}}$. These bounds give necessary conditions for a strict gain. □

D.5 Optimal rollout width under affine expected costs

The general comparison above permits arbitrary positive $c_e(N)$. An explicit accounting model is $c_e(N) = h_e + d_e N$, where $h_e \ge 0$ is expected block setup cost and $d_e > 0$ is expected cost per original rollout, including all associated work. For PWPO, d_e includes rejected certification attempts and probability-weighted additional likelihood work on accepted pairs. This affine model applies when the chosen cost unit is additive across rollouts. Nonlinear GPU batching or cross-rollout caching can be represented by the measured general function $c_e(N)$. The next proposition optimizes the exact leading coefficient Q_e.

Proposition 3 (Exact width selection)**.** *Suppress e and let $c(N) = h + dN$, $V(N) = H + a/N + b/[N(N-1)]$, with $h, H, a, b \ge 0$ and $d > 0$. The discrete increment of $Q(N) = c(N)V(N)$ is*

$$Q(N+1) - Q(N) = dH - \frac{ha}{N(N+1)} - \frac{db}{N(N-1)} - \frac{2hb}{N(N^2-1)}. \tag{D.43}$$

If $H > 0$, a minimizing integer is the smallest $N \ge 2$ for which this increment is nonnegative. It is unique unless that increment is zero, in which case N and $N+1$ are both minimizers. If $H = 0$ and $ha + b > 0$, Q strictly decreases to da and has no finite minimizer over all widths. If $H = 0$ and $ha = b = 0$, $Q(N) = da$ for every width.

Proof. Expanding the product gives

$$\begin{aligned} Q(N) &= (h + dN)\left(H + \frac{a}{N} + \frac{b}{N(N-1)}\right) \\ &= hH + da + dHN + \frac{ha}{N} + \frac{db}{N-1} + \frac{hb}{N(N-1)}. \end{aligned} \tag{D.44}$$

Taking differences, $1/(N+1)-1/N=-1/[N(N+1)]$ and $1/N-1/(N-1)=-1/[N(N-1)]$. For the last term, the exact cancellation is

$$\frac{1}{N(N+1)}-\frac{1}{N(N-1)}=\frac{N-1-(N+1)}{N(N-1)(N+1)}=-\frac{2}{N(N^2-1)}. \tag{D.45}$$

This proves Equation (D.43). Each subtracted nonnegative term decreases with N. Thus the increment is nondecreasing, strictly increasing if $ha+b>0$, and converges to dH. If $H>0$, it eventually becomes positive; before its first nonnegative value Q decreases, and afterwards Q increases. A zero increment joins exactly two minimizing adjacent integers in the strictly increasing case. When $ha=b=0$ and $H>0$, the increment is the positive constant dH, so the unique minimizer is 2. If $H=0$, Equation (D.43) is strictly negative precisely when $ha+b>0$, and Equation (D.44) gives the limit da. The remaining constant case is immediate from the same expansion.

One can also locate the minimizer through a strictly convex real relaxation. For $n>1$, direct differentiation gives

$$\begin{aligned} Q'(n)&=dH-\frac{ha}{n^2}-\frac{db}{(n-1)^2}-\frac{hb(2n-1)}{n^2(n-1)^2},\\ Q''(n)&=\frac{2ha}{n^3}+\frac{2db}{(n-1)^3}+\frac{2hb(3n^2-3n+1)}{n^3(n-1)^3}\geq 0. \end{aligned} \tag{D.46}$$

All numerators in the second line are nonnegative, and $3n^2-3n+1>0$. When $ha+b>0$, strict convexity shows that $Q'(2)\geq 0$ places the real minimizer at 2; otherwise, for $H>0$, the unique root $n_\star>2$ of $Q'(n)=0$ determines the integer optimum by comparing the objective at $\lfloor n_\star\rfloor$ and $\lceil n_\star\rceil$. □

For a finite budget, the affordable widths satisfy $2\leq N\leq\lfloor(\mathcal{B}-h)/d\rfloor$. The optimal finite-budget risk is obtained by evaluating the exact expression in Equation (D.25) over these integers. The floor in its denominator may shift the minimizer relative to Proposition 3. The sequence and PWPO widths are optimized separately.

D.6 Sharp coverage-dependent variance envelope

The exact gain depends jointly on pair acceptance, score separation, and reward variation. We now quantify the strongest possible gain from a given acceptance probability under bounded scores. This result is for one fixed task, so $H=0$, and gives a constructive sharpness check for the constants in the LOO analysis.

Proposition 4 (Sharp fixed-task gain envelope)**.** *Suppose R lies in an interval of width $W>0$ and $\|\psi_\theta(\tau)\|\leq G$ on the support at the fixed policy. Let $p=\Pr(|C|=2\mid x)$, counting distinct accepted pairs only. For $t_N=1/(N-1)$, define*

$$z_N(p)=\min\left\{p,\frac{1+pt_N}{2(2-p+pt_N)}\right\},\qquad \Phi_N(p)=(1+pt_N)z_N(p)-(2-p+pt_N)z_N(p)^2. \tag{D.47}$$

The conditional trace reduction $\Delta_N(x)$ satisfies

$$0\leq\Delta_N(x)\leq\frac{W^2G^2}{N}\Phi_N(p). \tag{D.48}$$

For every $N\geq 2$ and $p\in[0,1]$, the upper bound is attainable in a scalar-score finite model with a fixed reward-preserving cell partition. There is no strictly positive universal lower bound based on p alone, even when $p=1$.

Proof. Write $\mathcal{A}=\{|C|=2\}$. Positivity and the norm bound first give

$$\begin{aligned} 0\leq\omega(c,x)&=\mathbf{1}_{\mathcal{A}}w_0w_1\|\psi_0-\psi_1\|^2\\ &\leq\mathbf{1}_{\mathcal{A}}w_0w_1(\|\psi_0\|+\|\psi_1\|)^2\\ &\leq 4G^2\mathbf{1}_{\mathcal{A}}w_0w_1\leq G^2\mathbf{1}_{\mathcal{A}}, \end{aligned} \tag{D.49}$$

where $4w_0w_1 \le (w_0+w_1)^2 = 1$. Normalize the reward to $r \in [0,1]$, write $\mu = \mathbb{E}r$, and let $v = \mathrm{Var}(r)$. The exact gain in Equation (D.22) then implies

$$\begin{aligned}\Delta_N(x) &= \frac{W^2}{N}\mathbb{E}\big[((r-\mu)^2 + t_N v)\omega\big] \\ &\le \frac{W^2G^2}{N}\left\{\mathbb{E}[\mathbf{1}_{\mathcal{A}}(r-\mu)^2] + t_N p v\right\}.\end{aligned} \tag{D.50}$$

If $p = 0$, both sides vanish. Otherwise put $t = t_N \in (0,1]$ and $u = \mathbb{E}[r \mid \mathcal{A}]$. Reflecting $r \mapsto 1-r$ preserves both centered moments, so assume $\mu \le 1/2$. Since $r^2 \le r$, $\mathbb{E}[\mathbf{1}_{\mathcal{A}}r^2] \le pu$ and $v \le \mu(1-\mu)$. Moreover, $0 \le u \le \min\{1, \mu/p\}$. Hence the quantity in braces, denoted F, admits the chain

$$\begin{aligned}F &= \mathbb{E}[\mathbf{1}_{\mathcal{A}}r^2] - 2\mu p u + p\mu^2 + tp(\mathbb{E}r^2 - \mu^2) \\ &\le pu - 2\mu p u + p\mu^2 + tp(\mu - \mu^2) \\ &= pu(1-2\mu) + p(1-t)\mu^2 + tp\mu \\ &\le p\min\{1, \mu/p\}(1-2\mu) + p(1-t)\mu^2 + tp\mu.\end{aligned} \tag{D.51}$$

The last inequality uses $1 - 2\mu \ge 0$.

When $0 \le \mu \le p$, the last line is the concave quadratic

$$\begin{aligned}F \le f(\mu) &:= (1+pt)\mu - (2-p+pt)\mu^2, \\ f(\mu) &= \frac{(1+pt)^2}{4(2-p+pt)} - (2-p+pt)\left(\mu - \frac{1+pt}{2(2-p+pt)}\right)^2.\end{aligned} \tag{D.52}$$

Its unconstrained maximizer is at most $1/2$, because $1 + pt \le 2 - p + pt$ for $p \le 1$. On $[0, \min\{p, 1/2\}]$ it is therefore maximized at $z_N(p)$. In the remaining possible interval $p \le \mu \le 1/2$, the upper bound in Equation (D.51) is

$$\begin{aligned}F &\le q(\mu) := p\{1 + (t-2)\mu + (1-t)\mu^2\}, \\ q'(\mu) &= p\{t - 2 + 2(1-t)\mu\} \le p\{t - 2 + (1-t)\} = -p < 0, \\ q(\mu) &\le q(p) = f(p) \le f(z_N(p)) = \Phi_N(p).\end{aligned} \tag{D.53}$$

Combining both intervals gives $F \le \Phi_N(p)$ and proves Equation (D.48). The bound retains the joint dependence between the event $\mathcal{A}$ and the centered reward.

To prove attainability, let $z = z_N(p)$. Use a reward-W two-member cell of total probability z, a reward-0 two-member cell of total probability $p - z$, and a reward-0 singleton of probability $1 - p$. Omit zero-mass cells. Within each pair, assign equal member probabilities and scalar scores $+G, -G$; the singleton score is zero. A differentiable family realizing these scores at $\theta = 0$ is

$$p_\theta(u) = \frac{p_0(u)\exp(\theta s(u))}{\sum_v p_0(v)\exp(\theta s(v))}, \qquad s(u) \in \{-G, 0, G\}, \qquad \partial_\theta \log p_\theta(u)|_{\theta=0} = s(u), \tag{D.54}$$

because $\mathbb{E}_{p_0} s = 0$. The normalized reward is Bernoulli with mean z, $\omega = G^2\mathbf{1}_{\mathcal{A}}$, and all inequalities involving $r^2 \le r$ are equalities. Its event-weighted moment is

$$\begin{aligned}\mathbb{E}[\mathbf{1}_{\mathcal{A}}(r-z)^2] + tp\,\mathrm{Var}(r) &= z(1-z)^2 + (p-z)z^2 + tpz(1-z) \\ &= (1+pt)z - (2-p+pt)z^2 = \Phi_N(p).\end{aligned} \tag{D.55}$$

The partition is fixed and rewards agree within each pair; it can be represented by transcripts with one certified binary-order choice. At the policy considered, every score satisfies the required bound. This realizes equality. For the absence of a positive lower bound, choose any distribution over cells and parameterize each member probability as half its cell mass. Both members then have identical scores, so $\omega = 0$ even if every cell is a pair. This completes both sharpness claims. □

D.7 Expected budgets versus deterministic hard caps

Theorem 3 gives an expected-cost bound with a deterministic sample count. A hard cap additionally requires a deterministic bound on each block's realized cost, including replay and scoring. Translating finite token/call limits into a wall-clock bound also requires a bound on the time per operation.

Corollary 2 (Hard cap with a deterministic per-block envelope)**.** *Suppose $C_{e,N} \le c_e^{\max}(N)$ almost surely for a known deterministic envelope. Prespecify $K_e^{\max} = \lfloor \mathcal{B}/c_e^{\max}(N) \rfloor \ge 1$ and execute all these blocks. Then*

$$\sum_{k=1}^{K_e^{\max}} C_{e,N}^{(k)} \le \mathcal{B} \quad \textit{almost surely}, \qquad \mathbb{E}\|\widehat{g}_{e,K_e^{\max},N} - g\|^2 = \frac{V_e(N)}{K_e^{\max}}. \tag{D.56}$$

Proof. The pathwise cost chain is

$$\sum_{k=1}^{K_e^{\max}} C_{e,N}^{(k)} \le \sum_{k=1}^{K_e^{\max}} c_e^{\max}(N) = c_e^{\max}(N) \left\lfloor \frac{\mathcal{B}}{c_e^{\max}(N)} \right\rfloor \le \mathcal{B}. \tag{D.57}$$

The deterministic sample count gives the risk identity by Theorem 2. Replacing mean costs by envelope costs gives the corresponding finite-budget comparisons for this conservative procedure. □

To see why a random denominator needs a separate analysis, let independent scalar block estimates G_k take values $+1, -1$ with equal probability and have true mean zero. Give them respective costs $1, 2$, and stop after the cumulative cost first reaches or exceeds 2. The stopped average has expectation

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} G_k\right] = \tfrac{1}{2}(-1) + \tfrac{1}{4}(1) + \tfrac{1}{4}\left(\frac{1-1}{2}\right) = -\tfrac{1}{4} \ne 0. \tag{D.58}$$

This stopping rule can overshoot the threshold, and correlation between cost and gradient biases the stopped average. Cost-dependent block selection and random sample counts require a separate analysis that accounts for this dependence and distinguishes $1/\mathbb{E}K$ from $\mathbb{E}[1/K]$.

D.8 Finite-compute optimization with fresh on-policy updates

Following the smooth stochastic-optimization framework of Ghadimi & Lan (2013), we derive how statistical efficiency enters a nonconvex convergence bound. The analyzed variant collects a fresh on-policy batch and takes one unpreconditioned stochastic-gradient ascent step per iteration.

Lemma 4 (Smooth ascent with the exact variance term)**.** *Let J have L-Lipschitz gradient, $L > 0$. Let $\mathcal{F}_t$ denote all information before the fresh batch at iteration t, and suppose θ_t is $\mathcal{F}_t$-measurable. If $\mathbb{E}[G_t \mid \mathcal{F}_t] = \nabla J(\theta_t)$ and $\mathbb{E}[\|G_t - \nabla J(\theta_t)\|^2 \mid \mathcal{F}_t] \le \bar{V}$, then for $\theta_{t+1} = \theta_t + \eta G_t$,*

$$\mathbb{E}[J(\theta_{t+1}) \mid \mathcal{F}_t] \ge J(\theta_t) + \eta(1 - L\eta/2)\|\nabla J(\theta_t)\|^2 - \frac{L\eta^2\bar{V}}{2}. \tag{D.59}$$

Proof. For any d, the fundamental theorem of calculus, Cauchy–Schwarz, and Lipschitz continuity give the step-by-step smoothness bound

$$\begin{aligned} J(\theta + d) - J(\theta) - \langle \nabla J(\theta), d\rangle &= \int_0^1 \langle \nabla J(\theta + ud) - \nabla J(\theta), d\rangle \, du \\ &\ge -\int_0^1 \|\nabla J(\theta + ud) - \nabla J(\theta)\| \, \|d\| \, du \\ &\ge -\int_0^1 Lu\|d\|^2 \, du = -\frac{L}{2}\|d\|^2. \end{aligned} \tag{D.60}$$

Set $d = \eta G_t$ and condition on $\mathcal{F}_t$. Unbiasedness eliminates the mean-noise cross term, so

$$\begin{aligned} \mathbb{E}[\|G_t\|^2 \mid \mathcal{F}_t] &= \|\nabla J(\theta_t)\|^2 + 2\,\langle \nabla J(\theta_t), \mathbb{E}[G_t - \nabla J(\theta_t) \mid \mathcal{F}_t]\rangle \\ &\quad + \mathbb{E}[\|G_t - \nabla J(\theta_t)\|^2 \mid \mathcal{F}_t] \\ &\le \|\nabla J(\theta_t)\|^2 + \bar{V}. \end{aligned} \tag{D.61}$$

Substitution into Equation (D.60) proves Equation (D.59). □

Corollary 3 (Finite compute and an exactly optimized stepsize certificate)**.** *Assume the conditions of Lemma 4, $J(\theta) \le J^\star < \infty$, and $\Delta = J^\star - J(\theta_0) > 0$. Suppose each iteration has conditional expected cost at most $\bar{c} > 0$. Prespecify $T_{\mathrm{opt}} = \lfloor \mathcal{B}/\bar{c} \rfloor \ge 1$, and choose I independently and uniformly from $\{0, \dots, T_{\mathrm{opt}} - 1\}$. For $0 < \eta \le 1/L$,*

$$\mathbb{E}\|\nabla J(\theta_I)\|^2 \le \frac{2\Delta/(T_{\mathrm{opt}}\eta) + L\eta\bar{V}}{2 - L\eta}. \tag{D.62}$$

Set $a = L\Delta/T_{\mathrm{opt}} > 0$. The right side is minimized exactly over these constant stepsizes by

$$\eta_\star = \frac{2a}{L\{a + \sqrt{a^2 + 2a\bar{V}}\}}, \qquad \mathbb{E}\|\nabla J(\theta_I)\|^2 \le a + \sqrt{a^2 + 2a\bar{V}}. \tag{D.63}$$

The expected total cost is at most $\mathcal{B}$.

Proof. Take total expectations in Equation (D.59) and sum over iterations. Since $\eta(1 - L\eta/2) > 0$, telescoping and the upper bound on J give

$$\begin{aligned}
\eta(1 - L\eta/2) \sum_{t=0}^{T_{\mathrm{opt}}-1} \mathbb{E}\|\nabla J(\theta_t)\|^2 &\le \sum_{t=0}^{T_{\mathrm{opt}}-1} \{\mathbb{E}J(\theta_{t+1}) - \mathbb{E}J(\theta_t)\} + \frac{L\eta^2\bar{V}T_{\mathrm{opt}}}{2} \\
&= \mathbb{E}J(\theta_{T_{\mathrm{opt}}}) - J(\theta_0) + \frac{L\eta^2\bar{V}T_{\mathrm{opt}}}{2} \\
&\le J^\star - J(\theta_0) + \frac{L\eta^2\bar{V}T_{\mathrm{opt}}}{2} = \Delta + \frac{L\eta^2\bar{V}T_{\mathrm{opt}}}{2}.
\end{aligned} \tag{D.64}$$

Divide by $T_{\mathrm{opt}}\eta(1 - L\eta/2)$ and use the uniform independent choice of I to prove Equation (D.62).

To optimize the bound exactly, write $s = L\eta \in (0, 1]$, $\beta = a + \sqrt{a^2 + 2a\bar{V}}$, and $s_\star = 2a/\beta$. Then $\beta \ge 2a > 0$, so $s_\star \in (0, 1]$, and $\beta^2 = 2a(\bar{V} + \beta)$. Completing the square gives an exact optimality certificate:

$$\begin{aligned}
\frac{2a/s + s\bar{V}}{2 - s} - \beta &= \frac{2a - 2\beta s + (\bar{V} + \beta)s^2}{s(2 - s)} \\
&= \frac{(\bar{V} + \beta)\{s - \beta/(\bar{V} + \beta)\}^2}{s(2 - s)} \\
&= \frac{(\bar{V} + \beta)(s - s_\star)^2}{s(2 - s)} \ge 0.
\end{aligned} \tag{D.65}$$

Equality holds at $s = s_\star$, proving Equation (D.63). If $\bar{V} = 0$, this yields $\eta_\star = 1/L$ and bound $2a$. For the remaining case $\Delta = 0$, θ_0 already attains the global upper bound and has zero gradient, so it can be returned directly.

For the budget statement, let C_t be the iteration cost. The tower property supplies

$$\mathbb{E}\sum_{t=0}^{T_{\mathrm{opt}}-1} C_t = \sum_{t=0}^{T_{\mathrm{opt}}-1} \mathbb{E}[\mathbb{E}(C_t \mid \mathcal{F}_t)] \le T_{\mathrm{opt}}\bar{c} = \bar{c}\left\lfloor \frac{\mathcal{B}}{\bar{c}} \right\rfloor \le \mathcal{B}. \tag{D.66}$$

The conditional assumptions allow correlated costs and gradients, as well as dependence between successive iterates. □

For a fresh average of K task blocks of width N, Theorem 2 gives the conditional variance $V_e(\theta_t, N)/K$. A uniform bound on that quantity supplies $\bar{V}$, while a uniform bound on the complete update cost supplies $\bar{c}$. Comparing training paths requires these bounds to hold uniformly along each path. The optimized certificate additionally satisfies

$$\sqrt{2a\bar{V}} \le a + \sqrt{a^2 + 2a\bar{V}} \le 2a + \sqrt{2a\bar{V}}, \qquad a = \frac{L\Delta}{\lfloor \mathcal{B}/\bar{c} \rfloor}. \tag{D.67}$$

For positive $\bar{V}$ and large budgets, its stochastic term is governed by $\bar{c}\bar{V}$; the deterministic term also depends on $\bar{c}$ separately. Equation (D.63) optimizes this smoothness-based stationarity bound for fresh on-policy updates under uniform cost and variance assumptions.