WHEN DOES BRANCHING PAY? AN EXACT VARIANCE ANALYSIS OF TREE-STRUCTURED POLICY GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative policy gradient methods sample a fixed number of independent completions per prompt, so the rollouts share nothing but the prompt. A growing literature makes use of tree-structured policy gradient estimators instead, branching at chosen points guided by heuristics. We formalize such designs as a pair (β, α): a kernel β that decides where and how much to branch, and an anchor map α that decides which ancestor supplies each edge's baseline group. For any such pair, we give an exactly unbiased on-policy gradient estimator and, to our knowledge, the first exact variance decomposition for tree policy gradients with adaptive branching and group baselines. From it, we derive design principles for where to branch and how to form baseline groups, and identify regimes in which branching does not help. We test these predictions on reasoning and multi-turn agentic tool-use tasks with verifiable rewards.

1 INTRODUCTION

Group-relative policy gradients (Shao et al., 2024; Ahmadian et al., 2024; Liu et al., 2025) hard-code a sampling design with G independent completions per prompt. A growing literature replaces this "star" structure with trees (Ji et al., 2026; Hou et al., 2025; Yang et al., 2025; Kazemnejad et al., 2025; Guo et al., 2025; Dong et al., 2025b;a), where the strategy is to branch rollouts at chosen intermediate points so that siblings share a prefix. Despite reported gains from tree-structured advantage estimation, independent group sampling remains prominent in published large-scale post-training recipes (GLM-5-Team et al., 2026; DeepSeek-AI et al., 2026). It remains unclear how to choose where to branch, by how much, or whether these methods are worthwhile compared to star-shaped baselines at the same sampling budget.

We provide a formal framework that views these tree-structured methods as combining a forward on-policy sampling algorithm with a backward advantage estimator that uses group baselines. We construct a tree policy gradient estimator that generalizes several existing methods and allows us to identify their biases. We then give an unbiased estimator, together with an *exact variance decomposition formula*, which applies to REINFORCE Leave-One-Out (RLOO) (Kool et al., 2019; Ahmadian et al., 2024) as a special case. To the best of our knowledge, this is the first variance analysis of the tree policy gradients themselves, rather than the advantage estimates Wang et al. (2026); He et al. (2026).

Reading the variance formula gives concrete design principles: we should

1. Prioritize branching at decision points that come before "noisy" segments of trajectories
2. Compute advantage baselines using more "local" comparison groups of trajectories only when these groups contain enough samples

However, implementing either principle in practice is non-trivial. In particular, it is a challenge to locate the decision points to branch at. High-entropy tokens have been identified empirically as important forks in reasoning (Wang et al., 2025), motivating entropy-guided branching (Hou et al., 2025; Dong et al., 2025b; He et al., 2026). Yet policy uncertainty need not coincide with policy gradient "noise": a high-entropy choice between equivalent actions may leave the outcome unchanged, while a stochastic tool response can determine success or failure even when the policy is

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.

confident. Entropy-guided branching can therefore spend samples on inconsequential locations and miss the decisions that matter.

The variance formula also identifies when branching is not worth it. Informally, if the *gradient noise stays flat or decreases along the rollout*, and baselines use the full group of trajectories, then branching later in the rollout cannot remove much variance. Earlier branching saves less computation and approaches independent sampling. We quantify this trade-off by bounding the possible variance reduction at the same expected sampling cost.

Figure 1 illustrates why the gradient noise profile matters for branching. Each rollout contains one stochastic action that determines success; all other actions are fixed. Gradient noise is therefore concentrated at a single step. When this noise occurs at the start, branching afterward adds no independent samples of the noisy step. Independent completions are therefore optimal. When the noise occurs at the end, sharing the preceding fixed steps allows more independent samples of the noisy step, so the tree reduces gradient variance.

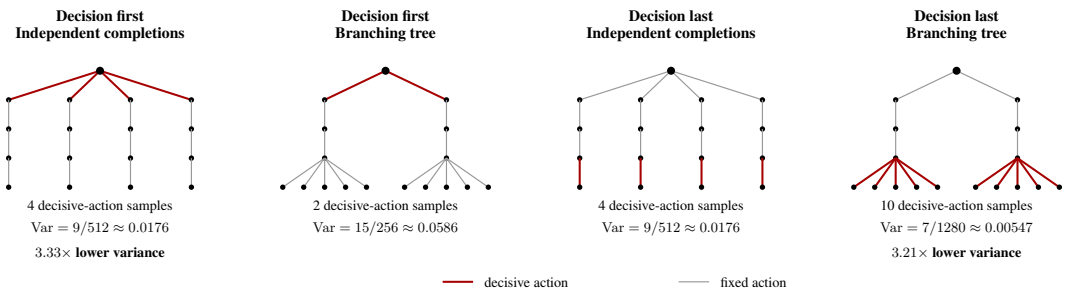

Figure 1: **Branching benefits depend on the gradient noise profile.** At equal sampling cost, noise concentrated at the start favors independent completions (left pair), while noise concentrated at the end favors branching (right pair). Appendix A gives the full construction and exact variance calculations.

We test these claims empirically with the `Qwen2.5-3B base` model (Qwen et al., 2025) throughout, on a single-turn task, MATH (Hendrycks et al., 2021), and on a multi-turn task, question answering with a retrieval tool on HotpotQA (Yang et al., 2018).

First, we measure where gradient noise resides along a rollout: on MATH most of it is concentrated in the early steps, whereas on HotpotQA most of it comes after the first tool-call, suggesting that **trees are better suited for multi-turn tasks than single-turn reasoning tasks**.

Second, we compare branching rules by the variance of the policy gradient estimate they produce, which we measure directly from repeated sampling. On both MATH and HotpotQA, **neither random nor entropy-triggered branching reduces the variance at matched cost by more than 10%**. However, a **better branching rule is theoretically possible**: an oracle rule that branches where our variance decomposition formula predicts the largest reduction does far better in both settings, validating our theory, although this rule relies on inputs not available at sampling time.

Finally, we train our base model on HotpotQA with the same sampling budget, using Tree-GRPO (Ji et al., 2026)'s random branching and ARPO (Dong et al., 2025b)'s entropy-triggered branching, each with its shipped advantage estimator and with our unbiased one. Neither consistently improves accuracy. This is in line with our measurements from the previous experiment, since neither rule substantially reduces the gradient variance.

2 Setup

We begin with a version of the ReAct (Yao et al., 2023b) framework, where an agent interacts with the environment to solve a given task. Suppose we have a policy π_θ, with terminal reward r at horizon L. For any state s, write the value function as $V(s) = \mathbb{E}_\theta[r \mid s]$, the policy gradient as $G(s) = \nabla_\theta V(s)$, and the (step-wise) policy scores as $\psi(s, a) = \nabla_\theta \log \pi_\theta(a \mid s)$. We provide an account of a forward sampling algorithm and a backward advantage estimator below. Exact details of the setup are given in Appendix B.

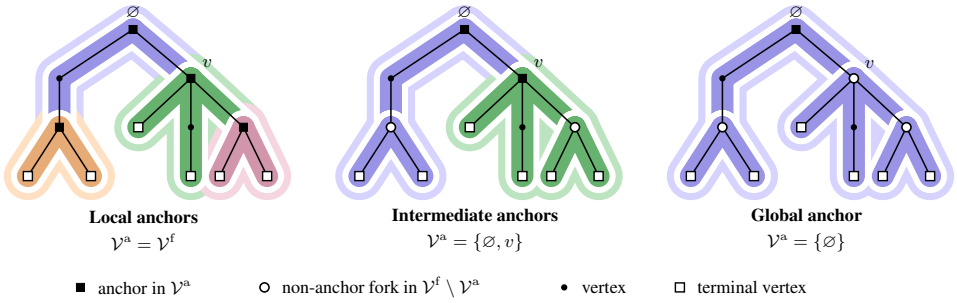

Figure 2: **Forks and anchors.** White boundaries separate the fork regions: each begins at a fork and contains its outgoing chains up to the next fork or a terminal vertex. Colors separate the anchor regions, whose edges use the same anchor for their advantage baseline group. The panels show local, intermediate, and global choices of anchors.

Tree sampling with branching kernel β For a given prompt x, we generate a random tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ of training data with observations $\{o_v\}_{v\in\mathcal{V}}$ attached to vertices, and actions $\{a_e\}_{e\in\mathcal{E}}$ attached to edges. For each vertex v, there is a unique ancestral path $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$ from the root $v_0 = \varnothing$ to the vertex itself $v_n = v$. We write the interaction history of the agent up to v as

$$s_v = (x, a_{e_1}, o_{v_1}, \cdots, a_{e_n}, o_v)$$

and the score of an edge $e = (v, u)$ as $\psi(e) = \psi(s_v, a_e)$, the score of the action taken at v. The number of outgoing edges from the vertex, N_v, is drawn from a *branching kernel* $\beta(\cdot \mid s_v)$. A vertex is considered *terminal* if it follows a terminal action, or its ancestral path has reached a maximum length L. Terminal vertices have no children. The root has at least two children.

Value estimation with vertex weights w Attach to each vertex $v \in \mathcal{V}$ the weight

$$w(v) = \prod_{u \prec v} \frac{1}{N_u}, \qquad w(\varnothing) = 1, \tag{1}$$

the probability of reaching v from the root by choosing a uniformly random child at every step. We can then define the value estimate at a vertex $v \in \mathcal{V}$ as

$$\widehat{V}_v = \sum_{\ell \succeq v,\ \ell \text{ terminal}} \frac{w(\ell)}{w(v)} r(s_\ell) \tag{2}$$

Group advantage estimation with anchor map α Given a sampled tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, denote the set of *forks*, or vertices in the tree with more than one child as $\mathcal{V}^{\mathrm{f}} \subseteq \mathcal{V}$. The set of *anchors* $\mathcal{V}^{\mathrm{a}}$ is a subset of $\mathcal{V}^{\mathrm{f}}$ that contains the root. For any vertex v, the anchor $\alpha(v)$ is its most recent ancestor in the anchor set, including itself. A *fork region* consists of a fork $f \in \mathcal{V}^{\mathrm{f}}$ and its N_f outgoing chains, each ending at the next fork or terminal history, while an *anchor region* consists of an anchor $v \in \mathcal{V}^{\mathrm{a}}$ and all vertices in $\alpha^{-1}(v)$ (see Figure 2). For an edge $e = (v, u)$, let

$$\rho_v = \frac{w(v)}{N_v w(\alpha(v))}, \qquad \widehat{A}(e) = \frac{\widehat{V}_u - \widehat{V}_{\alpha(v)}}{1 - \rho_v} \tag{3}$$

We can think of ρ_v as the fraction of the anchor $\alpha(v)$'s mass that flows through e, and show that $\widehat{A}(e) = \widehat{V}_u - b(e)$, where $b(e)$ is the value estimate of the anchor computed from the terminal vertices which are not below e. This generalizes the REINFORCE leave-one-out estimator (Kool et al., 2019; Ahmadian et al., 2024), which has a single fork and anchor region.

Dropping importance ratios, PPO clipping and regularisation, we define the policy gradient estimate with respect to the branching kernel and the anchor map to be

$$\widehat{g}_{\beta,\alpha} = \sum_{e=(v,u)\in\mathcal{E}} w(u)\widehat{A}(e)\psi(e). \tag{4}$$

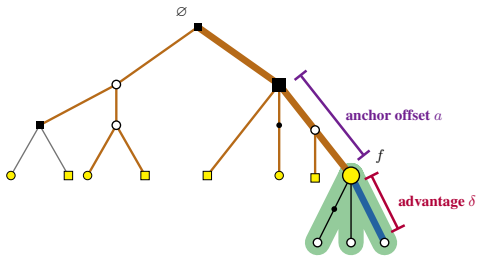

Figure 3: **Variance terms at fork f.** The green band marks the fork region at f, to be revealed next. The yellow vertices mark the frontier of the revealed subtree so far. Policy scores along orange edges are combined into the normalized prefix score ϕ; policy scores along the blue edge make up the chain score Ψ. The purple and red line intervals represent the anchor offset and the advantage respectively, which are value differences between the corresponding nodes.

We write the anchor map through its anchor set: $\widehat{g}_{\beta,\varnothing}$ takes the root as the only anchor, and $\widehat{g}_{\beta,\mathcal{V}^{\mathrm{f}}}$ makes every fork an anchor, so that each edge is baselined at its last branch point. Let β_k give the root k children and every other vertex one, so that it samples k *independent completions*; then $\hat{g}_{\star,k} = \widehat{g}_{\beta_k,\varnothing}$ is the REINFORCE leave-one-out estimator. We write β_{rand} for the kernel of Tree-GRPO (Ji et al., 2026), which expands randomly chosen vertices of each tree, and β_{ent} for that of ARPO (Dong et al., 2025b), which forks a trajectory where the token entropy rises after a tool call.

3 Main Results

3.1 Unbiasedness

The baseline $b(e)$ of an edge is built from trajectories that avoid e, so $b(e)\psi(e)$ averages to zero as long as $b(e)$ carries no information about the action on e. This fails if the sampler looks ahead: a rule that adds siblings after a continuation has failed makes the baseline depend on that continuation. Previsibility rules this out.

Assumption 3.1 (Previsibility)**.** *For any vertex v, its number of children N_v and the event $\{v \in \mathcal{V}^{\mathrm{a}}\}$ must be path-predictable, meaning that branching and anchor set membership can only depend on the observed information on the ancestral path $\varnothing \to v$, and not retrospectively on the resulting descendant observations or rewards.*

Theorem 3.2 (Unbiasedness)**.** *Under Assumption 3.1, for every (β, α),*

$$\mathbb{E}[\widehat{g}_{\beta,\alpha}] = G(s_\varnothing) \tag{5}$$

The proof is Theorem E.2, under the regularity conditions of Assumption B.1.

Note that neither branching kernel shipped with our baseline methods, Tree-GRPO and ARPO, is previsible. Tree-GRPO picks its expansion vertices after each seed chain has terminated, and ARPO shares a fixed rollout budget among a prompt's chains and restarts unused budget at the prompt. The resulting bias is not measured.

3.2 Exact Variance Decomposition

In order to decompose the variance of the policy gradient estimator $\widehat{g}_{\beta,\alpha}$, note that any sampled tree $\mathcal{T}$ can be partitioned into fork regions. So we can enumerate these regions in any valid topological order, and add up the variance contribution of each fork region, conditional on the revealed subtree so far. Looking at a subtree $\mathcal{T}' \subset \mathcal{T}$, we refer to its terminal vertices as the *frontier* $\mathcal{L}$, and note that each vertex on the frontier is either a fork or terminal in $\mathcal{T}$. Let $f \in \mathcal{V}^{\mathrm{f}} \cap \mathcal{L}$ be a fork on this frontier with state $s_f = s$.

Chain advantage and score Suppose it takes T steps[1] under the branching kernel β to get from f to the next fork with state S_T, so we have a *chain* $(s, S_1, \cdots, S_T)$ with edges $e_1, \cdots, e_T$. Define

$$\delta = V(S_T) - V(s), \qquad \Psi = \sum_{t=1}^{T} \psi(e_t) \tag{6}$$

which we can think of as the cumulative advantage and policy score on the chain. We can think of $\delta\Psi + G(S_T) - G(s)$ as the irreducible gradient error introduced by this chain.

Anchor offset Suppose now that f has $v = \alpha(f)$ as its anchor. By using the leave-one-out value estimate of s_v as the chain's advantage baseline instead of the true value at s_f, we introduce an *anchor offset* a_f that is fixed before the chain at f is revealed.

$$a_f = \frac{V(s) - \widehat{V}_v^{\mathcal{L}}}{1 - \rho_f}, \qquad \widehat{V}_v^{\mathcal{L}} = \sum_{u \in \mathcal{L}, v \preceq u} \frac{w(u)}{w(v)} V(s_u) \tag{7}$$

We can think of $a_f\Psi$ as the gradient error introduced by the anchor offset.

Normalized prefix score Next, we look at the cumulative policy score on the path $\varnothing \to f$, that has been centered relative to the scores in all the revealed anchor regions that the path passes through

$$\phi_f = \sum_{\substack{e=(u,v)\in\mathcal{T}' \\ v \preceq f}} \psi(e) - \sum_{\substack{e=(u,v)\in\mathcal{T}' \\ v \not\preceq f,\ \alpha(u) \preceq f}} \frac{\rho_u}{1-\rho_u} \psi(e). \tag{8}$$

We can think of $\delta\phi$ as the gradient error introduced by the chain's value innovation to earlier decisions and competing branches.

Definition 3.3 (Gradient noise)**.** The gradient noise at s, given prefix score ϕ and anchor offset a is

$$C(s; \phi, a) = \mathbb{E}_s \left\| \varepsilon(\phi, a) \right\|^2, \qquad \varepsilon(\phi, a) = \delta(\Psi + \phi) + a\Psi + G(S_T) - G(s) \tag{9}$$

Note that C is not just reward variance, and persists even with exact values and gradients. We can also show that $\mathbb{E}_s\delta = \mathbb{E}_s\Psi = 0$ and $\mathbb{E}_s[\delta\Psi] = G(s) - \mathbb{E}_s G(S_T)$, hence $\mathbb{E}_s\varepsilon(\phi, a) = 0$.

Definition 3.4 (Sibling noise)**.** For two independent copies of chains at f, with advantages δ_1, δ_2 and scores Ψ_1, Ψ_2, define the sibling noise as

$$B(s) = \mathbb{E}_s \left\| \delta_1\Psi_2 + \delta_2\Psi_1 \right\|^2 = 2\mathbb{E}_s[\delta^2]\mathbb{E}_s \left\| \Psi \right\|^2 + 2 \left\| \mathbb{E}_s[\delta\Psi] \right\|^2 \tag{10}$$

This is the additional noise arising from estimating the baseline with other chains.

Theorem 3.5 (Exact gradient variance)**.** *For the estimator equation 4,*[2]

$$\boxed{\mathrm{Var}(\widehat{g}_{\beta,\alpha}) = \mathbb{E} \sum_{f \in \mathcal{V}^{\mathrm{f}}} \frac{w(f)^2}{N_f} \left[C(s_f; \phi_f, a_f) + \frac{(N_f - 1)\rho_f^2}{2(1-\rho_f)^2} B(s_f) \right]} \tag{11}$$

The proof is Theorem F.3 of the appendix. It reveals the fork regions one at a time in a fixed order (Lemma F.2); the prefix score and anchor offset depend on that order, the variance itself does not.

4 Design Implications

4.1 Cost

A tree can always lower variance by sampling more, so we compare estimators only at equal cost.

Definition 4.1 (Cost)**.** The cost of a sampled tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is its number of edges, $\mathrm{cost}(\mathcal{T}) = |\mathcal{E}|$. The cost of a branching kernel β is the expected cost of the trees it samples, $\mathrm{cost}(\beta) = \mathbb{E}_\beta[\mathrm{cost}(\mathcal{T})]$.

[1] T itself is a random variable known as a stopping time.

[2] The variance of a vector is its expected squared Euclidean deviation, $\mathrm{Var}(\widehat{g}) = \mathbb{E} \left\| \widehat{g} - \mathbb{E}\widehat{g} \right\|^2$, the trace of its covariance.

For example, k independent completions that run to the horizon have $\mathrm{cost}(\beta_k) = kL$.

Definition 4.2 (Variance at matched cost)**.** Let β be a kernel whose cost lies between that of N and $N+1$ independent completions, $\mathrm{cost}(\beta_N) \le \mathrm{cost}(\beta) \le \mathrm{cost}(\beta_{N+1})$ with $N \ge 2$, and let

$$\lambda = \frac{\mathrm{cost}(\beta) - \mathrm{cost}(\beta_N)}{\mathrm{cost}(\beta_{N+1}) - \mathrm{cost}(\beta_N)} \in [0, 1].$$

Sampling $N+1$ independent completions with probability λ and N otherwise costs $\mathrm{cost}(\beta)$ in expectation; we say it has *matched cost*, and call its variance

$$\mathrm{Var}_\star(\beta) = (1-\lambda)\mathrm{Var}(\hat{g}_{\star,N}) + \lambda \mathrm{Var}(\hat{g}_{\star,N+1})$$

the *matched variance*. An estimator $\widehat{g}$ on trees sampled by β has *relative variance* $\mathrm{Var}(\widehat{g})/\mathrm{Var}_\star(\beta)$, or, if it is biased, *relative mean squared error* $\mathbb{E}\left\|\widehat{g} - G(s_\varnothing)\right\|^2 / \mathrm{Var}_\star(\beta)$. A value below one means that β improves on independent completions at the same cost.

4.2 Where to branch

Giving a vertex v m children instead of one multiplies the cost below v by m. By Theorem 3.5, it pays when the gradient noise $C(s_v)$ it averages is large compared with the sibling noise $B(s_v)$ it adds.

Proposition 4.3 (informal)**.** *If the fork inflates the gradient noise C at and below v by at most a factor κ, and its sibling term, the $B(s_v)$ term of Theorem 3.5, is at most $\eta\, C(s_v)$, then*

$$\mathrm{Var}_v(\textit{fork}) \le \frac{\kappa + \eta}{m}\, \mathrm{Var}_v(\textit{no fork}),$$

where Var_v conditions on everything outside the subtree of v (Appendix G).

> **Branching Location Principle**: Prioritize branching before large gradient noise C.

4.3 How to form baseline groups

Making a fork f an anchor sets its anchor offset a_f to zero, removing the $a_f \Psi$ part of the gradient noise C, but its children are then baselined against only their $N_f - 1$ siblings, which raises the sibling-noise B term of Theorem 3.5.

Proposition 4.4 (informal)**.** *If the offset does not happen to cancel chain noise, re-anchoring at f lowers its fork-region contribution to the variance when*

$$a_f^2 > \frac{2\,\mathbb{E}_{s_f}[\delta^2]}{N_f - 1},$$

with δ the chain advantage of Section 3.2 (Appendix H).

> **Anchor Selection Principle**: Prioritize re-anchoring when the anchor offset a is large, especially when continuation values vary little or the number of branches is large.

4.4 When do trees help

A fork can only average the gradient noise still unresolved where it is placed. To locate that noise along a rollout, take n independent completions that run to the horizon L, and let $\mathcal{F}_t$ be the information in the first t actions of each, with their observations. The remaining variance and the gradient-noise profile are

$$r_t = \mathbb{E}\big[\mathrm{Var}(\hat{g}_{\star,n} \mid \mathcal{F}_t)\big], \qquad q_t = n\,(r_{t-1} - r_t), \qquad t = 1, \dots, L, \tag{12}$$

so that $r_0 = \mathrm{Var}(\hat{g}_{\star,n})$, $r_L = 0$, and $q_t \ge 0$ is the variance revealed by the t-th action, scaled so that $\sum_t q_t = n\mathrm{Var}(\hat{g}_{\star,n})$.

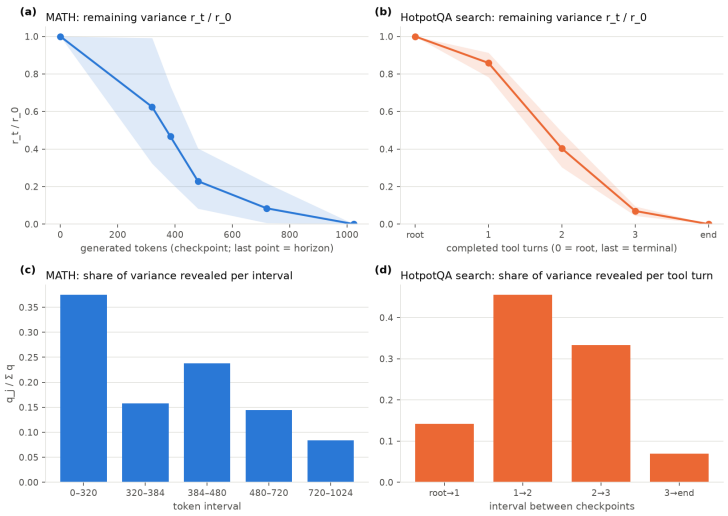

Figure 4: Gradient-noise profiles on MATH (left) and HotpotQA (right), with $n = 4$. Top: remaining variance r_t/r_0, with 95% bootstrap confidence intervals over prompts. Bottom: share of the variance revealed in each interval.

Proposition 4.5 (informal)**.** *If the profile is front-loaded, $q_1 \geq \cdots \geq q_L$, a tree that waits to branch cannot beat independent completions at the same cost: if it has n root children, costs as much as m completions, and first forks below the root only after a fraction n/m of the horizon, then whatever its anchors, its variance is at least that of m independent completions (Appendix I).*

The noise revealed before the first fork cannot be averaged by any fork, and under a front-loaded profile it is already too large for the tree to catch up.

When trees do not help: When the gradient noise is revealed early, branch early or not at all.

5 Experiments

5.1 Gradient noise profiles

Setup. We estimate r_t and q_t of equation 12 for $n = 4$ by resampling every rollout's continuation from checkpoint t, on 16 prompts per task, at temperature 1. On MATH the policy is Qwen2.5-3B after 20 GRPO steps, the prompts have pass rates in $(0.1, 0.9)$, and the checkpoints are at 320, 384, 480 and 720 generated tokens. On HotpotQA it is Qwen2.5-3B after 60 steps of search training, with up to three searches per rollout and checkpoints after each completed search (Appendix K).

Results. On MATH, 38% of the variance is revealed in the first 320 tokens (Figure 4), so by Section 4.4 a fork has to come early to help. On HotpotQA the first tool turn reveals only 14% and the next two 46% and 33%, leaving 86% open after the first search.

Trees better suited for multi-turn tasks: Multi-turn HotpotQA search has a much more favorable gradient noise profile than single-turn MATH for tree policy gradient estimators.

5.2 Where to branch

Setup. We ask where a single fork should go. Each tree has four chains from the prompt, one of which forks once at a candidate vertex: after a completed search on HotpotQA, or at one of four fixed token positions on MATH. The fork has 2, 3 or 4 children (the columns of Table 1). We

		children per fork		
	placement rule	2	3	4
MATH	random vertex	1.02 [0.98, 1.05]	1.12 [1.06, 1.21]	1.24 [1.13, 1.39]
	entropy rise (ARPO)	1.04 [0.98, 1.12]	1.15 [1.04, 1.30]	1.21 [1.10, 1.37]
	theory rule (oracle inputs)	**0.86** [0.82, 0.90]	—	—
HotpotQA	random vertex	0.96 [0.92, 0.99]	1.00 [0.96, 1.04]	1.07 [1.02, 1.11]
	entropy rise (ARPO)	0.91 [0.86, 0.94]	0.96 [0.89, 1.02]	1.03 [0.94, 1.11]
	theory rule (oracle inputs)	**0.63** [0.58, 0.68]	—	—

Table 1: Variance of a single fork relative to independent completions at matched cost, by placement rule and children per fork: mean over 26 prompts with 95% bootstrap intervals (per prompt in Appendix J.3). The theory rule is an oracle and is evaluated at two children only.

report the variance of $\widehat{g}_{\beta,\varnothing}$ over 64 replicates divided by that of independent completions at the same expected cost (Definition 4.2); below one, the fork helps. Three rules choose the vertex: a random candidate, the largest entropy rise (ARPO's trigger), and a theory rule that minimises the fork's term in Theorem 3.5. The theory rule is an oracle, not a deployable procedure (Appendix J).

Results. Random and entropy forks stay close to independent completions, and entropy is not detectably better than random (sign test $p = 0.42$). The theory rule cuts the variance substantially on both tasks (Table 1), and its predicted variance change matches the realised one (Appendix J.2).

Optimal branching points are missed opportunities: Branching at the right points significantly reduces variance at matched cost, but existing heuristics fail to identify them.

Entropy is not enough to guide branching: Entropy is not enough for predicting optimal branching locations, and in fact is not detectably better than random.

5.3 Multi-turn training at equal sampling budget

Setup. We ask whether branching raises accuracy in training. We train Qwen2.5-3B for 180 steps on HotpotQA search under three kernels at approximately matched sampling budget per step: independent completions, Tree-GRPO's β_{rand} and ARPO's β_{ent}. Each is trained with its shipped z-scored advantage and with our estimators $\widehat{g}_{\beta,\varnothing}$ and $\widehat{g}_{\beta,\mathcal{V}^{\mathrm{f}}}$ (the rows of Table 2), three seeds each, and scored greedily on HotpotQA, 2WikiMultiHopQA, MuSiQue and Bamboogle. Neither shipped kernel is previsible (Section 3.1), so the bias of our estimators on them is unmeasured (Appendix L). ARPO is taken out of the setting it was published in: we run its branching kernel in Tree-GRPO's environment and reward, from the base model, without its cold-start SFT on agentic trajectories, its Python tool or its reward bonus for using several tools.

Results. Every seed that trains without failing lands in the same accuracy band, whatever the kernel or estimator. The rows differ mainly in how many seeds fail, but three seeds per row cannot rank them on this. This matches Section 5.2: at matched cost the heuristics' forks barely reduce the variance, so training has little to gain from them.

Collapse. Eight of the 24 runs collapse or never learn, among them seeds of the shipped GRPO and ARPO, so a single training run can land far from its row's typical score. Note that the collapse of GRPO-style training in agentic RL is a known failure (Deng et al., 2026), and neither the Tree-GRPO nor the ARPO paper reports results over multiple seeds.

No variance reduction, no training gain: In our particular setup, where we have already shown that random and entropy-triggered branching yield only modest variance reductions at the same budget, we see no gain in test accuracy from them either.

	Avg. EM at step 180				sampling per step	
estimator	seed 1	seed 2	seed 3	healthy mean	calls	tokens
GRPO (shipped)	34.0	33.2	$0.2^{\dagger}$	33.6	2,427	1.55 M
$\hat{g}_{\star,4}$	$22.2^{\dagger}$	34.0	$0.0^{\dagger}$	34.0	2,342	1.52 M
Tree-GRPO (shipped)	34.9	35.0	34.5	34.8	2,570	1.63 M
$\widehat{g}_{\beta_{\mathrm{rand}},\varnothing}$	32.2	35.7	28.9	32.3	2,486	1.57 M
$\widehat{g}_{\beta_{\mathrm{rand}},\mathcal{V}^{\mathrm{f}}}$	$5.1^{\dagger}$	$1.6^{\dagger\ddagger}$	$16.8^{\dagger}$	—	—	—
ARPO (shipped)	$31.5^{\ddagger}$	$17.2^{\dagger}$	$16.9^{\dagger\ddagger}$	—	—	—
$\widehat{g}_{\beta_{\mathrm{ent}},\varnothing}$	34.5	30.2	34.2	33.0	2,224	1.53 M
$\widehat{g}_{\beta_{\mathrm{ent}},\mathcal{V}^{\mathrm{f}}}$	34.2	34.5	$32.0^{\ddagger}$	34.4	2,827	1.87 M

Table 2: Trained on HotpotQA. Avg. EM is the exact-match score (0–100) averaged over HotpotQA, 2WikiMultiHopQA, MuSiQue and Bamboogle at step 180 (per dataset in Table J.2); sampling per training step is over the healthy seeds (tool calls and sampled plus retrieved tokens; — where a row has none). † failed to train, ‡ gradient freeze (Appendix L.2); the healthy mean and the sampling columns are over unmarked seeds.

6 Related work

Group-relative estimators and values. GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), DAPO (Yu et al., 2025) and Dr. GRPO (Liu et al., 2025) sample independent completions with the root as the only anchor; without normalisation the estimator is $\hat{g}_{\star,k}$. Our $\widehat{V}$ is a Monte Carlo value, as in TRPO's vine estimator (Schulman et al., 2017) and VinePPO (Kazemnejad et al., 2025), but computed on the training tree itself rather than from extra rollouts or a GAE critic (Schulman et al., 2018).

Tree sampling. ARPO (Dong et al., 2025b), AEPO (Dong et al., 2025a), Tree-GRPO (Ji et al., 2026) and BranPO (Zhao et al., 2026) for agents, and TreeRL (Hou et al., 2025), TreeRPO (Yang et al., 2025), TreePO (Li et al., 2025) and SPO (Guo et al., 2025) for single-turn reasoning, are choices of β and α: they fork at high-entropy points, at random or at fixed intervals, and baseline against the root, the fork, or both; GiGPO (Feng et al., 2025) instead groups matching states of independent completions. We show that placement decides the variance, that entropy is a poor guide to it, and that the shipped ARPO and Tree-GRPO kernels are not previsible. MCTS pipelines (Zhang et al., 2024; Chen et al., 2024) and inference-time search (Yao et al., 2023a; Hao et al., 2023) branch on value estimates of the partial tree, so their data are off-policy.

Splitting and concurrent work. Weighted ensemble sampling (Aristoff, 2018) splits Markov-chain trajectories with the same weight division as our w, is unbiased when bins are chosen from the past, and allocates replicas by a variance analysis. Concurrent EPIG-Tree (Khomich et al., 2026) allocates branches by expected information gain about the gradient per unit of compute. We instead give the exact variance for any previsible kernel and anchor map, bound the gain by the gradient-noise profile, and test at matched cost on multi-turn search.

7 Conclusion

We cast tree-structured policy gradients as a forward sampling kernel paired with a backward anchored advantage, gave an unbiased estimator and an exact decomposition of its variance, and derived from it where to branch, how to form baseline groups, and when branching cannot beat independent completions at the same cost. On MATH and HotpotQA, where a tree branches matters more than whether it does: random and entropy-triggered forks barely change the variance at matched cost, while the placements the decomposition identifies remove a large share of it. One limitation of our work is that the branching placement rule reads its inputs from extra rollout continuations, so it measures what good placement can achieve rather than providing a sampler; estimating those inputs cheaply is the natural next step for any future work, and the decomposition says which quantities to estimate.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, to find related work, to execute experiments, to draft sections of the paper, help develop theoretical models and to verify mathematical proofs. We have not used generative AI tools for generating synthetic datasets, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work, including checking LLM-generated research ideas for potential plagiarism through manual literature surveys, and manually reviewing all polished theoretical content in paper drafts to ensure it complies with its original meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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, 2024. URL https://arxiv.org/abs/2402.14740.

David Aristoff. Analysis and optimization of weighted ensemble sampling. *ESAIM: Mathematical Modelling and Numerical Analysis*, 52(4):1219–1238, 2018. ISSN 1290-3841. doi: 10.1051/m2an/2017046. URL http://dx.doi.org/10.1051/m2an/2017046.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: Process supervision without process, 2024. URL https://arxiv.org/abs/2405.03553.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng,

Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. On group relative policy optimization collapse in agent search: The lazy likelihood-displacement, 2026. URL https://arxiv.org/abs/2512.04220.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization, 2025a. URL https://arxiv.org/abs/2510.14545.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization, 2025b. URL https://arxiv.org/abs/2507.19849.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models, 2025. URL https://arxiv.org/abs/2505.23564.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model, 2023. URL https://arxiv.org/abs/2305.14992.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning, 2026. URL https://arxiv.org/abs/2607.14171.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search, 2025. URL https://arxiv.org/abs/2506.11902.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning, 2026. URL https://arxiv.org/abs/2509.21240.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms, 2025. URL https://arxiv.org/abs/2410.01679.

Nikita Khomich, Leopold Hermansson, and Ido Hakimi. Epig-tree: Compute-optimal branching for gradient-efficient reinforcement learning, 2026. URL https://arxiv.org/abs/2609.20004.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free!, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, Zheng Zhang, Wei Shen, Qian Liu, Chenghua Lin, Jian Yang, Ge Zhang, and Wenhao Huang. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling, 2025. URL https://arxiv.org/abs/2508.17445.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017. URL https://arxiv.org/abs/1502.05477.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

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.

Hongcheng Wang, Yinuo Huang, Sukai Wang, Guanghui Ren, and Hao Dong. Why tree-style branching matters for thought advantage estimation in grpo, 2026. URL https://arxiv.org/abs/2509.24494.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. Treerpo: Tree relative policy optimization, 2025. URL https://arxiv.org/abs/2506.05183.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search, 2024. URL https://arxiv.org/abs/2406.03816.

Yubao Zhao, Weiquan Huang, Sudong Wang, Ruochen Zhao, Chen Chen, Yao Shu, and Chengwei Qin. Branpo: Scalable contrastive branch sampling for long-horizon agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2602.03719.

A MOTIVATING EXAMPLE

Every completion has horizon $L \geq 2$. Only step d is random: $A \sim \text{Bernoulli}(p)$ with $p = (1 + e^{-\theta})^{-1}$, reward $r = A$ and score $A - p$. All other scores vanish, so $G(s_\varnothing) = p(1-p)$ and $\hat{g}_{\star,n} = \sum_i (A_i - \bar{A})^2/(n-1)$. Binomial moments give

$$F_p(n) := \text{Var}(\hat{g}_{\star,n}) = \frac{p(1-p)(1-2p)^2}{n} + \frac{2p^2(1-p)^2}{n(n-1)},$$

which is decreasing and convex for $n \geq 2$.

Early noise ($d = 1$). A root-anchored tree with N root children gives $\hat{g}_{\star,N}$. Each child costs at least L, so expected cost at most mL implies $\mathbb{E}N \leq m$. Jensen's inequality yields $\text{Var}(\widehat{g}_{\beta,\varnothing}) = \mathbb{E}F_p(N) \geq F_p(m)$.

Late noise ($d = L$). Continue b root children to depth $L-1$, then give each k terminal children. The cost is $b(L-1) + bk$. With root anchoring, the bk independent outcomes have weight and mass ratio $1/(bk)$, giving variance $F_p(bk)$.

For Figure 1, $L = 4$, $p = 1/4$, $b = 2$, $k = 5$. Both the tree and four independent completions cost 16. Their early-noise variances are $15/256$ and $9/512$ (a $10/3$ ratio); their late-noise variances are $7/1280$ and $9/512$ (a $45/14$ improvement). More generally, $b = 2$, $k = L+1$ costs $4L$ and improves late-noise variance over four completions by asymptotically $3L/4$ at $p = 1/4$.

B FINITE-HORIZON INTERACTION AND POLICY GRADIENTS

B.1 HISTORIES, TERMINATION, AND REGULARITY

Let $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{O}$ be standard Borel spaces of prompts, actions, and observations. Set

$$\mathcal{S}_t = \mathcal{X} \times (\mathcal{A} \times \mathcal{O})^t, \qquad \mathcal{S} = \bigcup_{t \geq 0} \mathcal{S}_t.$$

Write $s \preceq s'$ when s is a prefix of s', and $|s| = t$ for $s \in \mathcal{S}_t$. Fix a horizon $L \geq 1$ and a termination action $\square \in \mathcal{A}$. The live and terminal histories are

$$\mathcal{S}^\circ = \{s \in \mathcal{S}_{<L} : \text{no action in } s \text{ equals } \square\}, \tag{13}$$

$$\partial\mathcal{S} = \{s \oplus (a, o) : s \in \mathcal{S}^\circ,\ a = \square \text{ or } |s| = L - 1\}. \tag{14}$$

The observation after $\square$ can be a fixed dummy symbol. The boundary is prefix-free, and every history of length L has a terminal prefix. Put $\overline{\mathcal{S}} = \mathcal{S}^\circ \cup \partial\mathcal{S}$.

A policy $\pi_\theta(da \mid s)$, with θ in an open subset of $\mathbb{R}^d$, selects actions. An environment kernel $P(do \mid s, a)$ selects observations. Both P and the terminal reward $r : \partial\mathcal{S} \to \mathbb{R}$ are independent of θ. For live histories, write

$$(M_\theta f)(s) = \int f(s \oplus (a, o))\, \pi_\theta(da \mid s) P(do \mid s, a). \tag{15}$$

Starting from $S_0 = x$, the interaction ends at

$$\tau = \min\{L, \inf\{t \geq 1 : A_t = \square\}\}.$$

When starting at a history of length k, time below refers to additional transitions and the remaining horizon is at most $L - k$.

Assumption B.1. *The reward satisfies $|r| \leq R < \infty$. The policy admits densities with respect to a fixed dominating measure, with locally parameter-independent support and sufficient differentiability and domination to differentiate the one-step and finite-horizon expectations below. Its score, as in Section 2,*

$$\psi(s, a) = \nabla_\theta \log \pi_\theta(a \mid s), \qquad \textit{written also } \psi(s \oplus (a, o)) = \psi(s, a),$$

is defined almost surely and satisfies, at the parameter under consideration,

$$\sup_{s \in \mathcal{S}^\circ} \int \|\nabla_\theta \log \pi_\theta(a \mid s)\|^2\, \pi_\theta(da \mid s) \leq \sigma_\psi^2 < \infty. \tag{16}$$

For a finite action space, differentiating a finite sum is straightforward once the probabilities are differentiable. Discreteness alone implies neither differentiability nor finite summation for a countably infinite action space. Regularity at zero probabilities also requires care.

B.2 VALUES AND SCORES

Define the value and its gradient by

$$V(s) = \begin{cases} r(s), & s \in \partial\mathcal{S}, \\ (MV)(s), & s \in \mathcal{S}^{\circ}, \end{cases} \qquad G(s) = \nabla_\theta V(s). \tag{17}$$

Backward recursion uniquely determines V, and $G = 0$ on $\partial\mathcal{S}$. If an extension to histories after termination is desired, set $V(s) = r(s_\partial)$ for the unique terminal prefix s_∂. This makes V harmonic even for a formal chain continued after termination. We use only pretermination scores in what follows.

Lemma B.2 (Score and gradient identities)**.** *For a live history s and an admissible function f,*

$$\nabla_\theta(M_\theta f)(s) = M_\theta(f\psi)(s) + M_\theta(\nabla_\theta f)(s), \tag{18}$$

$$(M\psi)(s) = 0, \qquad G(s) = M(G + V\psi)(s). \tag{19}$$

Consequently, with the chain started at s,

$$G(s) = \mathbb{E}_s\left[\sum_{t=1}^{\tau} V(S_t)\psi(S_t)\right], \qquad \|G(s)\| \le R(L - |s|)\sigma_\psi. \tag{20}$$

The score of the terminating action is included.

Proof. Differentiate equation 15 and use $\nabla\pi_\theta = \pi_\theta \nabla \log \pi_\theta$. Differentiating total probability gives $M\psi = 0$. Applying the first identity to V gives equation 19. Iterating until termination gives equation 20; conditional Cauchy–Schwarz gives the bound. Equivalently, the stopped processes

$$V(S_{t\wedge\tau}), \qquad G(S_{t\wedge\tau}) + \sum_{j=1}^{t\wedge\tau} V(S_j)\psi(S_j)$$

are martingales. □

A baseline measurable before an action is sampled has zero expected product with that action's score. Below, an edge's baseline may also depend on independent competing branches.

C TREE SAMPLING AND ANCHORED ADVANTAGES

C.1 A HISTORY-DEPENDENT BRANCHING KERNEL

Use Ulam–Harris labels $\mathcal{U} = \bigcup_{n\ge 0} \mathbb{N}^n$, where $\mathbb{N} = \{1, 2, \ldots\}$ and the root is $\varnothing$. A finite ordered rooted tree has vertex set $\mathcal{V} \subset \mathcal{U}$, child counts N_v, and children $v1, \ldots, vN_v$. Write $\mathcal{V}^{\mathrm{f}} = \{v : N_v \ge 2\}$ and $\mathcal{V}^{\mathrm{l}} = \{v : N_v = 0\}$. For an edge $e = (v, u)$ as in Section 2, write $e^- = v$ and $e^+ = u$ for its parent and child.

Fix a prompt x and write $s_\varnothing = x$ for the root history. At a vertex with history s_v, independently draw $N_v \sim \beta(\cdot \mid s_v)$ and then draw N_v independent action–observation pairs from $\pi_\theta(da \mid s_v)P(do \mid s_v, a)$. The kernel may also read the child counts on the path to v, as Assumption 3.1 allows; nothing below uses more than that N_v is drawn before, and independently of, the actions leaving v. Fresh randomness is used at each vertex and edge. Assume

$$\beta(\{2, 3, \ldots\} \mid s_\varnothing) = 1, \qquad \beta(\{0\} \mid s) = \mathbf{1}\{s \in \partial\mathcal{S}\}. \tag{21}$$

Counts are finite almost surely. Every path has length at most L, so the whole tree is finite almost surely, even if its expected size is infinite. Leaves are exactly terminal histories. In this article β is a fixed sampling design; we do not differentiate a tree-sampling objective.

Define the uniform-descent mass

$$w(v) = \prod_{u \prec v} N_u^{-1}, \qquad w(\varnothing) = 1. \tag{22}$$

Conditional on the sampled tree, this is the probability that successive uniform child choices visit v. The value estimate is

$$\widehat{V}_v = \sum_{\substack{\ell \in \mathcal{V}^{\mathrm{l}} \\ v \preceq \ell}} \frac{w(\ell)}{w(v)} r(s_\ell). \tag{23}$$

Thus $\widehat{V}_\ell = r(s_\ell)$ and $\widehat{V}_v = N_v^{-1} \sum_i \widehat{V}_{vi}$ at internal vertices.

C.2 Previsible anchors and branch exclusion

Let $\mathcal{F}_v^{\mathrm{path}}$ denote the information on the path to v, including its history, child count, and existence. The anchor set $\mathcal{V}^{\mathrm{a}} \subseteq \mathcal{V}^{\mathrm{f}}$ contains the root and is previsible (Assumption 3.1):

$$\{v \in \mathcal{V}^{\mathrm{a}}\} \in \mathcal{F}_v^{\mathrm{path}}.$$

The anchor of a vertex and an edge is

$$\alpha(v) = \max_{\preceq} \{u \in \mathcal{V}^{\mathrm{a}} : u \preceq v\}, \qquad \alpha(e) = \alpha(e^-). \tag{24}$$

The root as the only anchor, $\mathcal{V}^{\mathrm{a}} = \{\varnothing\}$, gives $\widehat{g}_{\beta,\varnothing}$; every fork an anchor, $\mathcal{V}^{\mathrm{a}} = \mathcal{V}^{\mathrm{f}}$, gives $\widehat{g}_{\beta,\mathcal{V}^{\mathrm{f}}}$. For each edge define

$$\rho_{e^-} = \frac{w(e^-)}{N_{e^-}\, w(\alpha(e^-))} = \frac{w(e^+)}{w(\alpha(e))}, \tag{25}$$

$$b(e) = \frac{1}{1-\rho_{e^-}} \sum_{\substack{\ell \in \mathcal{V}^{\mathrm{l}},\ \alpha(e) \preceq \ell \\ e^+ \not\preceq \ell}} \frac{w(\ell)}{w(\alpha(e))} r(s_\ell), \tag{26}$$

$$\widehat{A}(e) = \widehat{V}_{e^+} - b(e), \qquad \psi(e) = \nabla_\theta \log \pi_\theta(a_e \mid s_{e^-}). \tag{27}$$

The path from an anchor to e^+ contains a fork, hence $0 < \rho_{e^-} \le 1/2$. The excluded mass is positive and $b(e)$ is a convex average. Splitting the anchor average gives

$$\widehat{A}(e) = \frac{\widehat{V}_{e^+} - \widehat{V}_{\alpha(e)}}{1-\rho_{e^-}}. \tag{28}$$

In particular, $|\widehat{V}_v|, |b(e)| \le R$ and $|\widehat{A}(e)| \le 2R$.

Definition C.1 (Tree policy-gradient estimator)**.** For $v \in \mathcal{V}$, set

$$\widehat{G}_v = \sum_{e:\, v \preceq e^-} \frac{w(e^+)}{w(v)} \widehat{A}(e)\psi(e), \qquad \widehat{g}_{\beta,\alpha} = \widehat{G}_\varnothing. \tag{29}$$

At the root this is the estimator of Section 2.

Although the sum defining $\widehat{G}_v$ uses edges below v, their baselines can use branches outside that subtree. The anchor rule must therefore be retained when studying subtree estimators.

D Stopping lines and truncated estimators

D.1 Information at a line

A line is an antichain of vertices. It cuts a subtree when every path from that subtree's root to a leaf meets the line exactly once; the singleton consisting of the subtree root is allowed. The frontier of Section 3.2 is such a cut. A random line $\mathcal{L}$ is a stopping line if

$$\{v \in \mathcal{L}\} \in \mathcal{F}_v^{\mathrm{path}} \quad \text{for every label } v.$$

For a line cutting the entire tree, $\mathcal{F}_{\mathcal{L}}$ is the information in the revealed tree on and above the line, including the line's vertices and their child counts, but not the edges leaving the line. Random

revealed sets are encoded by marked labels and their membership indicators. If $\mathcal{L} \preceq \mathcal{L}'$ means the first cut is met no later than the second on every path, then

$$\mathcal{F}_{\mathcal{L}} \subseteq \mathcal{F}_{\mathcal{L}'}.$$

For a fixed existing vertex v, let $\mathcal{H}_v$ be the exterior information: the data at v and at all vertices not strictly below v. This includes all other subtrees, unlike the path field $\mathcal{F}_v^{\mathrm{path}}$. Statements at fixed labels are understood on the event of their existence; when a target cut is used, they are restricted to labels on or above that cut.

Proposition D.1 (Branching property)**.** *Given $\mathcal{F}_{\mathcal{L}}$ at a stopping cut, all outgoing branches from its vertices are conditionally independent. At $u \in \mathcal{L}$, each branch is distributed as*

$$a \sim \pi_\theta(\cdot \mid s_u), \quad o \sim P(\cdot \mid s_u, a), \quad \textit{a fresh tree from } s_u \oplus (a, o).$$

The same statement holds for branches from u conditional on $\mathcal{H}_u$.

Proof. Realize the construction with mutually independent random seeds for child counts and for action–observation pairs. Expose the path data until reaching the cut. Whether a vertex is in a stopping line is decided without reading randomness strictly below it. Conditional on the revealed data, the unread seeds in its separate branches retain their independent original laws. Conditioning on the exterior of one fixed subtree has the same property. Standard Borel spaces allow this random-seed realization of the kernels. □

In particular, for a bounded variable B determined outside the entire branch rooted at ui (including its incoming edge),

$$\mathbb{E}[B\psi(e_{ui}) \mid \mathcal{H}_u] = 0. \tag{30}$$

The history and child count at u are already known in this conditional expectation.

D.2 Fork regions and anchor regions

For an anchor u, let $\mathcal{L}_u^\alpha$ consist of the first subsequent anchors or leaves on each descending path. The edges between u and this cut, including those entering the cut, are exactly

$$E_u^\alpha = \{e : \alpha(e) = u\}.$$

For a fork f, let $\mathcal{L}_f^\beta$ consist of the first subsequent forks or leaves, and let E_f^β be the intervening edges. This region comprises N_f chains from f to the next forks or leaves. Both are stopping cuts of their respective subtrees. Globally,

$$E = \bigsqcup_{u \in \mathcal{V}^{\mathrm{a}}} E_u^\alpha = \bigsqcup_{f \in \mathcal{V}^{\mathrm{f}}} E_f^\beta, \qquad E_u^\alpha = \bigsqcup_{f : \alpha(f) = u} E_f^\beta. \tag{31}$$

These statements follow by selecting the last anchor or last fork on an edge's ancestral path. This is a refinement of region partitions; it does not assert that equally numbered fork and anchor generations coincide.

D.3 Truncation at an arbitrary cut

Write $v \preceq \mathcal{L}$ when v lies on or above the cut. Conservation of mass gives

$$\sum_{\substack{u \in \mathcal{L} \\ v \preceq u}} \frac{w(u)}{w(v)} = 1. \tag{32}$$

For any cut, not necessarily a stopping cut, define

$$\widehat{V}_v^{\mathcal{L}} = \sum_{\substack{u \in \mathcal{L} \\ v \preceq u}} \frac{w(u)}{w(v)} V(s_u), \tag{33}$$

$$b^{\mathcal{L}}(e) = \frac{1}{1-\rho_{e^-}} \sum_{\substack{u \in \mathcal{L},\ \alpha(e) \preceq u \\ e^+ \not\preceq u}} \frac{w(u)}{w(\alpha(e))} V(s_u), \tag{34}$$

$$\widehat{A}^{\mathcal{L}}(e) = \widehat{V}_{e^+}^{\mathcal{L}} - b^{\mathcal{L}}(e) = \frac{\widehat{V}_{e^+}^{\mathcal{L}} - \widehat{V}_{\alpha(e)}^{\mathcal{L}}}{1-\rho_{e^-}}, \tag{35}$$

$$\widehat{G}_v^{\mathcal{L}} = \sum_{\substack{e:\, v \preceq e^- \\ e^+ \preceq \mathcal{L}}} \frac{w(e^+)}{w(v)} \widehat{A}^{\mathcal{L}}(e) \psi(e) + \sum_{\substack{u \in \mathcal{L} \\ v \preceq u}} \frac{w(u)}{w(v)} G(s_u). \tag{36}$$

If $v \in \mathcal{L}$, these are $V(s_v)$ and $G(s_v)$. At the leaf cut they are $\widehat{V}_v$ and $\widehat{G}_v$. For stopping cuts, the baseline $b^{\mathcal{L}}(e)$ is determined outside the branch rooted at e^+: membership in the cut for competing vertices uses only their own paths. The ratios ρ_{e^-} and the anchor are also known before the incoming edge is sampled.

Lemma D.2 (Square integrability)**.** *All the estimators above are square integrable. At the root there is a single square-integrable bound, independent of the cut, for* $\left\|\widehat{G}_{\varnothing}^{\mathcal{L}}\right\|$.

Proof. At any depth the sum of vertex masses is at most one. Define

$$Q = \sum_{e \in E} w(e^+) \left\|\psi(e)\right\|.$$

Weighted Cauchy–Schwarz yields

$$Q^2 \le L \sum_{e \in E} w(e^+) \left\|\psi(e)\right\|^2, \qquad \mathbb{E}Q^2 \le L^2 \sigma_\psi^2.$$

For the second inequality condition on each parent's history and child count before sampling its children, then sum over at most L depths. No expected tree-size bound is needed because the weights normalize each generation. From equation 36,

$$\left\|\widehat{G}_{\varnothing}^{\mathcal{L}}\right\| \le 2RQ + RL\sigma_\psi. \tag{37}$$

The same argument with relative masses $w(z)/w(v)$ works below a fixed vertex, also conditionally on $\mathcal{H}_v$. Value estimates are bounded by R. □

E Line increments and unbiasedness

E.1 An algebraic increment identity

Let $\mathcal{L} \preceq \mathcal{L}'$ be cuts. For $u \in \mathcal{L}$ define

$$D_u = \widehat{V}_u^{\mathcal{L}'} - V(s_u), \qquad H_u = \widehat{G}_u^{\mathcal{L}'} - G(s_u).$$

For $v \preceq \mathcal{L}$ and $u \in \mathcal{L}$, set

$$\phi_v^{\mathcal{L}}(u) = \sum_{\substack{e:\, v \preceq e^- \\ e^+ \preceq \mathcal{L}}} \frac{\psi(e)}{1-\rho_{e^-}} \big(\mathbf{1}\{e^+ \preceq u\} - \rho_{e^-} \mathbf{1}\{\alpha(e) \preceq u\} \big). \tag{38}$$

An on-path edge contributes $\psi(e)$. An off-path edge whose anchor is above u contributes $\psi(e)$ times $-\rho_{e^-}/(1-\rho_{e^-})$, and other edges contribute zero. This is the normalized prefix score of Section 3.2, a vector in $\mathbb{R}^d$.

Proposition E.1 (Cut increments). *For $v \preceq \mathcal{L}$,*

$$\widehat{V}_v^{\mathcal{L}'} - \widehat{V}_v^{\mathcal{L}} = \sum_{\substack{u \in \mathcal{L} \\ v \preceq u}} \frac{w(u)}{w(v)} D_u, \tag{39}$$

$$\widehat{G}_v^{\mathcal{L}'} - \widehat{G}_v^{\mathcal{L}} = \sum_{\substack{u \in \mathcal{L} \\ v \preceq u}} \frac{w(u)}{w(v)} H_u + \sum_{u \in \mathcal{L}} \frac{w(u)}{w(v)} D_u \phi_v^{\mathcal{L}}(u). \tag{40}$$

The second sum in equation 40 runs over the entire cut, including vertices outside the subtree of v.

Proof. Group the vertices of $\mathcal{L}'$ according to their ancestors on $\mathcal{L}$, using equation 32. This proves equation 39. Set $d_z = \widehat{V}_z^{\mathcal{L}'} - \widehat{V}_z^{\mathcal{L}}$. For old edges,

$$\widehat{A}^{\mathcal{L}'}(e) - \widehat{A}^{\mathcal{L}}(e) = \frac{d_{e^+} - d_{\alpha(e)}}{1 - \rho_{e^-}}.$$

Insert equation 39, multiply by $w(e^+)\psi(e)/w(v)$, and interchange sums. The coefficient of $w(u)D_u/w(v)$ is equation 38. New edges and new boundary terms group into the first sum in equation 40. □

If a single internal cut vertex u is replaced by its children, the local increments are

$$D_u = \frac{1}{N_u} \sum_i V(s_{ui}) - V(s_u), \tag{41}$$

$$H_u = \frac{1}{N_u} \sum_i \Big[G(s_{ui}) + \big(V(s_{ui}) - b^{\mathcal{L}'}(e_{ui})\big)\psi(e_{ui}) \Big] - G(s_u). \tag{42}$$

At an anchor u, the baseline in equation 42 is the leave-one-out average of the other children's values. Otherwise it also includes the other cut vertices below $\alpha(u)$.

E.2 A CONDITIONAL PROOF WITH A FIXED EXTERIOR CUT

Theorem E.2 (Conditional unbiasedness). *Under Assumption B.1 and equation 21, for a stopping cut $\mathcal{L}'$ and an existing vertex u on or above it,*

$$\mathbb{E}[\widehat{V}_u^{\mathcal{L}'} \mid \mathcal{H}_u] = V(s_u), \qquad \mathbb{E}[\widehat{G}_u^{\mathcal{L}'} \mid \mathcal{H}_u] = G(s_u). \tag{43}$$

In particular, $\mathbb{E}[\widehat{g}_{\beta,\alpha}] = G(s_\varnothing) = \nabla_\theta V(s_\varnothing)$.

Proof. Use backward induction in depth. At a target-cut vertex there is nothing to prove. Otherwise form the auxiliary cuts

$$\mathcal{C} = \{u\} \cup \{z \in \mathcal{L}' : u \not\preceq z\}, \qquad \mathcal{C}^+ = (\mathcal{C} \setminus \{u\}) \cup \mathrm{ch}(u).$$

They keep the exterior at its *target cut*, and satisfy $\mathcal{C} \preceq \mathcal{C}^+ \preceq \mathcal{L}'$ on the event under consideration. The exterior part is $\mathcal{H}_u$-measurable. These auxiliary cuts are used algebraically; no global stopping-line assertion about a random selection of u is needed.

The one-step value increment from $\mathcal{C}$ to $\mathcal{C}^+$ has conditional mean zero by $MV = V$. Its gradient increment at u has conditional mean zero by equation 42, equation 30, and equation 19. The baseline for each new edge excludes its entire branch, so this argument applies even though the exterior is frozen.

For the refinement from $\mathcal{C}^+$ to $\mathcal{L}'$, exterior vertices are already in the target cut and have zero local increments. Thus equation 40 reduces to

$$\widehat{G}_u^{\mathcal{L}'} - \widehat{G}_u^{\mathcal{C}^+} = \frac{1}{N_u} \sum_i \big[\widehat{G}_{ui}^{\mathcal{L}'} - G(s_{ui}) + (\widehat{V}_{ui}^{\mathcal{L}'} - V(s_{ui}))\phi_u^{\mathcal{C}^+}(ui) \big].$$

The coefficient $\phi_u^{\mathcal{C}^+}(ui)$ is $\mathcal{H}_{ui}$-measurable, and $\mathcal{H}_u \subseteq \mathcal{H}_{ui}$. Apply the induction hypothesis and the tower property. The value proof uses the corresponding reduction of equation 39. Square integrability from Lemma D.2 justifies the products and conditional expectations. At the root $\mathcal{H}_\varnothing$ includes $N_\varnothing$, which may be random; the conditional mean is nevertheless the fixed vector $G(s_\varnothing)$. □

Corollary E.3 (Line martingales)**.** *For nested stopping cuts $\mathcal{L} \preceq \mathcal{L}'$ and a vertex v on or above them,*

$$\mathbb{E}[\widehat{V}_v^{\mathcal{L}'} \mid \mathcal{F}_{\mathcal{L}}] = \widehat{V}_v^{\mathcal{L}}, \qquad \mathbb{E}[\widehat{G}_v^{\mathcal{L}'} \mid \mathcal{F}_{\mathcal{L}}] = \widehat{G}_v^{\mathcal{L}}.$$

Proof. For each $u \in \mathcal{L}$, $\mathcal{F}_{\mathcal{L}} \subseteq \mathcal{H}_u$ on the corresponding event. Theorem E.2 gives zero conditional means for D_u and H_u. The coefficients in equation 39–equation 40 are $\mathcal{F}_{\mathcal{L}}$-measurable. Take conditional expectations, interpreting sums over random vertices as sums over their countable labels. □

For completeness, let $\kappa_\alpha(v)$ count strict anchor ancestors and define padded anchor generations

$$\overline{\mathcal{L}}^{(k)} = \{u \in \mathcal{V}^{\mathrm{a}} : \kappa_\alpha(u) = k\} \cup \{\ell \in \mathcal{V}^{\mathrm{l}} : \kappa_\alpha(\ell) \le k\}.$$

Successive generations cut complete anchor regions. Conditional on the earlier cut, the contributions

$$\frac{w(u)}{w(v)}\left[H_u \mathbf{1}\{v \preceq u\} + D_u \phi_v^{\overline{\mathcal{L}}^{(k)}}(u)\right], \quad u \in \overline{\mathcal{L}}^{(k)} \cap \mathcal{V}^{\mathrm{a}},$$

are independent: each uses only the fresh branches within its own anchor region and coefficients already revealed. Individual fork contributions need not have this independence when several forks share an anchor.

F An exact variance decomposition

F.1 Adaptive revelation of fork regions

Fix a deterministic total order of labels. Begin with $\mathcal{L}_0 = \{\varnothing\}$. At stage k, select the first fork f_k on the current cut and replace it by $\mathcal{L}_{f_k}^{\beta}$:

$$\mathcal{L}_{k+1} = (\mathcal{L}_k \setminus \{f_k\}) \cup \mathcal{L}_{f_k}^{\beta}. \tag{44}$$

After $K = |\mathcal{V}^{\mathrm{f}}|$ expansions the cut consists of leaves. Keep it fixed thereafter. Let $\mathscr{F}_k$ be the information exposed by this algorithm, including all frontier child counts, and set $g_k = \widehat{G}_\varnothing^{\mathcal{L}_k}$. Then $g_0 = G(s_\varnothing)$ and $g_K = \widehat{g}_{\beta,\alpha}$.

These adaptive frontiers need not be path-defined stopping lines: choosing which fork to expand can use other branches. Nevertheless, the unexplored branches at f_k are conditionally independent with their original laws given $\mathscr{F}_k$. The algorithm has read no outgoing-edge or descendant randomness there. Separate independent seeds for a vertex's child count and for its outgoing edges make this statement immediate even though N_{f_k} is known.

F.2 The stopped-chain laws

Starting from a live history s, generate an ordinary trajectory, drawing a count $N_t \sim \beta(\cdot \mid S_t)$ at each reached history. Define

$$T = \inf\{t \ge 1 : N_t \ne 1\}, \quad \Psi = \sum_{t=1}^{T} \psi(e_t), \quad \delta = V(S_T) - V(s). \tag{45}$$

Here e_t is the chain's t-th edge, so $\psi(e_t) = \psi(S_t)$ in the notation of Assumption B.1, and $T \le \tau \le L - |s|$; the count at the starting history does not determine T. This is exactly the law of one outgoing chain in a fork region. For $\phi \in \mathbb{R}^d$ and $a \in \mathbb{R}$, define

$$\varepsilon(\phi, a) = \delta(\Psi + \phi) + a\Psi + G(S_T) - G(s), \tag{46}$$

$$\varepsilon_{\mathrm{sib}} = \delta_1 \Psi_2 + \delta_2 \Psi_1, \tag{47}$$

where subscripts denote two independent chains from the same s.

Lemma F.1 (Chain identities). *With expectation $\mathbb{E}_s$ under this stopped-chain law,*

$$\mathbb{E}_s\delta = 0, \qquad \mathbb{E}_s\Psi = 0, \tag{48}$$

$$\mathbb{E}_s \|\Psi\|^2 = \mathbb{E}_s \sum_{t=1}^{T} \|\psi(S_t)\|^2 \le (L - |s|)\sigma_\psi^2, \tag{49}$$

$$\mathbb{E}_s[\delta\Psi] = G(s) - \mathbb{E}_s G(S_T), \qquad \mathbb{E}_s\varepsilon(\phi, a) = 0, \tag{50}$$

$$B(s) := \mathbb{E}_s \|\varepsilon_{\mathrm{sib}}\|^2 = 2\mathbb{E}_s[\delta^2]\mathbb{E}_s \|\Psi\|^2 + 2 \|\mathbb{E}_s[\delta\Psi]\|^2 . \tag{51}$$

Proof. Use the martingales in Lemma B.2, augmented by the counts observed at reached states. The counts do not change the conditional law of the next action. Optional stopping applies because T is bounded. The stopped score increments are martingale differences, giving the score-energy identity. Furthermore, for each t, on $\{t \le T\}$ the conditional mean of $V(S_T)$ given the information at t is $V(S_t)$. Therefore

$$\mathbb{E}_s[V(S_T)\Psi] = \mathbb{E}_s \sum_{t=1}^{T} V(S_t)\psi(S_t) = G(s) - \mathbb{E}_s G(S_T).$$

Subtract $V(s)\mathbb{E}_s\Psi = 0$ to obtain equation 50. Finally expand the sibling square and use independence; the cross term is $2 \|\mathbb{E}_s[\delta\Psi]\|^2$. □

F.3 A SINGLE FORK-REGION INCREMENT

For $f = f_k$, every edge of E_f^β has the same anchor $\alpha(f)$ and mass ratio

$$\rho_f = \frac{w(f)}{N_f w(\alpha(f))} \le \frac{1}{N_f} \le \frac{1}{2}. \tag{52}$$

Define the normalized prefix score and anchor offset of Section 3.2 by

$$\phi_f = \phi_\varnothing^{\mathcal{L}_k}(f), \qquad a_f = \frac{V(s_f) - \widehat{V}_{\alpha(f)}^{\mathcal{L}_k}}{1 - \rho_f}. \tag{53}$$

Both are $\mathscr{F}_k$-measurable; ϕ_f is a vector and a_f a scalar. Let $\delta_{f,i}$, $\Psi_{f,i}$, and $G_{f,i}$ denote the innovation, accumulated score, and terminal gradient of chain i. Given $\mathscr{F}_k$, these triples are independent copies of the chain law from s_f.

Lemma F.2 (Orthogonal fork expansion). *Let $\varepsilon_{f,i} = \varepsilon_i(\phi_f, a_f)$ and $\varepsilon_{f,ij} = \delta_{f,i}\Psi_{f,j} + \delta_{f,j}\Psi_{f,i}$. Then*

$$g_{k+1} - g_k = \frac{w(f)}{N_f} \left[\sum_{i=1}^{N_f} \varepsilon_{f,i} - \frac{\rho_f}{1 - \rho_f} \sum_{i<j} \varepsilon_{f,ij} \right]. \tag{54}$$

All displayed summands have conditional mean zero and are pairwise conditionally orthogonal. Consequently,

$$\mathbb{E}[\|g_{k+1} - g_k\|^2 \mid \mathscr{F}_k] = \frac{w(f)^2}{N_f} \left[C(s_f; \phi_f, a_f) + \frac{(N_f - 1)\rho_f^2}{2(1 - \rho_f)^2} B(s_f) \right], \tag{55}$$

where $C(s; \phi, a) = \mathbb{E}_s \|\varepsilon(\phi, a)\|^2$.

Proof. Only f has a nonzero local increment in equation 40, so $g_{k+1} - g_k = w(f)(H_f + D_f\phi_f)$, with $D_f = N_f^{-1}\sum_i \delta_{f,i}$. The anchor value changes by

$$\widehat{V}_{\alpha(f)}^{\mathcal{L}_{k+1}} - \widehat{V}_{\alpha(f)}^{\mathcal{L}_k} = \rho_f \sum_i \delta_{f,i}.$$

Every edge on chain i has the new advantage

$$\frac{V(s_f) + \delta_{f,i} - \widehat{V}_{\alpha(f)}^{\mathcal{L}_{k+1}}}{1 - \rho_f} = a_f + \delta_{f,i} - \frac{\rho_f}{1 - \rho_f} \sum_{j \ne i} \delta_{f,j}.$$

Sum along chains and collect unordered pairs to obtain equation 54.

Lemma F.1 gives centering. Distinct single-chain terms are independent. In the product of a single-chain term with a sibling term, at least one of the sibling indices is a different chain and contributes a centered δ or Ψ. Two distinct sibling terms either have disjoint indices or share exactly one; in the latter case, expanding the product leaves a centered factor from an unshared chain. All cross expectations vanish. Count N_f single terms and $\binom{N_f}{2}$ pair terms to obtain equation 55. The same argument gives zero cross matrices, not just zero expected inner products. □

Theorem F.3 (Exact gradient variance)**.** *Under Assumption B.1 and equation 21,*

$$\boxed{\mathrm{Var}(\widehat{g}_{\beta,\alpha}) = \mathbb{E}\,\|\widehat{g}_{\beta,\alpha} - G(s_\varnothing)\|^2 = \mathbb{E}\sum_{f\in\mathcal{V}^{\mathrm{f}}} \frac{w(f)^2}{N_f}\left[C(s_f;\phi_f,a_f) + \frac{(N_f-1)\rho_f^2}{2(1-\rho_f)^2}B(s_f)\right].} \tag{56}$$

The first equality is Theorem E.2. The parameters ϕ_f, a_f refer to the exploration order equation 44.

Proof. Lemma F.2 makes $(g_k, \mathscr{F}_k)$ a martingale, stopped after K. Its increments are orthogonal, so for each deterministic n,

$$\mathbb{E}\,\|g_n - G(s_\varnothing)\|^2 = \sum_{k=0}^{n-1} \mathbb{E}\,\|g_{k+1} - g_k\|^2 .$$

Almost surely $K < \infty$, hence $g_n \to \widehat{g}_{\beta,\alpha}$. The square-integrable common bound equation 37 gives convergence in L^2, not merely pointwise convergence. Take the limit on the left and use monotone convergence on the nonnegative series on the right. Substitute equation 55; every fork is expanded once. □

The covariance-matrix version replaces squared norms by outer products. In particular, writing $m(s) = \mathbb{E}_s[\delta\Psi]$, the sibling matrix is

$$\mathbb{E}_s[\varepsilon_{\mathrm{sib}}\varepsilon_{\mathrm{sib}}^\top] = 2\mathbb{E}_s[\delta^2]\mathbb{E}_s[\Psi\Psi^\top] + 2m(s)m(s)^\top.$$

G WHERE TO BRANCH: PRECISE STATEMENT

Proposition G.1 (Branch before large gradient noise)**.** *At a non-root live vertex v, compare one continuation (β) with $m \geq 2$ independent continuations (β'), keeping the anchor map and continuation laws fixed. Condition on their common exterior data, including s_v but excluding the imposed child count; denote this conditional variance by Var_v. Write $C = C(s_v;\phi_v,a_v)$, $C' = C(s_v;\phi_v,a'_v)$ and $\rho_v = w(v)/(mw(\alpha(v)))$. Let R, R' be the expected sums of later fork-region contributions under the respective designs, with the same conditioning. If $\kappa \geq 1$, $\eta \geq 0$ and*

$$C' \leq \kappa C, \qquad \frac{(m-1)\rho_v^2}{2(1-\rho_v)^2}B(s_v) \leq \eta C, \qquad R' \leq \frac{\kappa}{m}R,$$

then

$$\mathrm{Var}_v(\widehat{g}_{\beta',\alpha}) \leq \frac{\kappa+\eta}{m}\mathrm{Var}_v(\widehat{g}_{\beta,\alpha}).$$

The reduction is strict if $\kappa + \eta < m$ and the original variance is positive. Expected subtree cost is multiplied by m.

Proof. Start the revelation of Theorem F.3 at the cut containing v and all exterior leaves. By Theorem E.2, the cut estimator is the conditional mean in either design; subsequent increments are conditionally orthogonal. The already revealed edges give the same prefix score ϕ_v. Lemma F.2 also applies with one child, when its pair sum is empty, so the first contribution is $w(v)^2C$ without the fork and at most $w(v)^2(\kappa+\eta)C/m$ with it. Thus

$$\begin{aligned}\mathrm{Var}_v(\widehat{g}_{\beta',\alpha}) &\leq \frac{\kappa+\eta}{m}w(v)^2C + \frac{\kappa}{m}R \\ &\leq \frac{\kappa+\eta}{m}\big[w(v)^2C + R\big] = \frac{\kappa+\eta}{m}\mathrm{Var}_v(\widehat{g}_{\beta,\alpha}).\end{aligned}$$

The cost statement follows from sampling m independent copies of the original continuation. □

H How to form baseline groups: precise statement

Proposition H.1 (When to re-anchor)**.** *Fix the sampling design and revealed context at a fork f with $N_f \geq 2$. Suppose $\mathbb{E}_{s_f} \|\Psi\|^2 > 0$ and*

$$C(s_f; \phi_f, a_f) - C(s_f; \phi_f, 0) \geq a_f^2 \mathbb{E}_{s_f} \|\Psi\|^2 .$$

Making f an anchor strictly reduces its conditional fork-region contribution equation 55 if

$$a_f^2 > \frac{2\mathbb{E}_{s_f}[\delta^2]}{N_f - 1}.$$

Proof. Re-anchoring leaves ϕ_f unchanged, sets $a_f = 0$ and changes ρ_f to $1/N_f$. The increase in the sibling penalty is at most $B(s_f)/[2(N_f - 1)]$. Since $B(s_f) \leq 4\mathbb{E}_{s_f}[\delta^2]\mathbb{E}_{s_f} \|\Psi\|^2$ by Cauchy–Schwarz, the contribution decreases by at least

$$\frac{w(f)^2}{N_f} \mathbb{E}_{s_f} \|\Psi\|^2 \left[a_f^2 - \frac{2\mathbb{E}_{s_f}[\delta^2]}{N_f - 1} \right] > 0.$$

□

This is a local guarantee under the stated no-cancellation condition; changes in other regions may offset the gain.

I Gradient-noise profiles bound late branching: proof

Proposition I.1. *Suppose every trajectory reaches horizon L and the profile in equation 12 satisfies $q_1 \geq \cdots \geq q_L$. Let β have $n \geq 2$ root children, child counts independent of sampled actions and observations, expected cost mL for an integer $m \geq n$, and no non-root fork before a depth $d \in \{1, \ldots, L\}$ with $d \geq nL/m$. For every admissible anchor map α,*

$$\mathrm{Var}_\star(\beta) = \mathrm{Var}(\hat{g}_{\star,m}) \leq \mathrm{Var}(\widehat{g}_{\beta,\alpha}).$$

Proof. Couple the first d steps with those of n independent completions. Both designs have n chains of weight $1/n$, anchored at the root through these steps. Conditional on $\mathcal{F}_d$, their continuation values have means $V(s_{i,d})$ and their remaining gradient contributions have means $G(s_{i,d})/n$, by Theorem E.2. Consequently

$$\mathbb{E}[\widehat{g}_{\beta,\alpha} \mid \mathcal{F}_d] = \mathbb{E}[\hat{g}_{\star,n} \mid \mathcal{F}_d].$$

Total variance and the non-increasing profile give

$$\mathrm{Var}(\widehat{g}_{\beta,\alpha}) \geq r_0 - r_d = \frac{1}{n} \sum_{t \leq d} q_t \geq \frac{d}{L} \mathrm{Var}(\hat{g}_{\star,n}).$$

For k independent completions, Theorem 3.5 gives

$$\mathrm{Var}(\hat{g}_{\star,k}) = \frac{C}{k} + \frac{B}{2k(k-1)},$$

where $C = C(s_\varnothing; 0, 0)$ and $B = B(s_\varnothing)$ use full completions and do not depend on k. Thus $k\mathrm{Var}(\hat{g}_{\star,k})$ is non-increasing, and

$$\mathrm{Var}(\widehat{g}_{\beta,\alpha}) \geq \frac{d}{L} \mathrm{Var}(\hat{g}_{\star,n}) \geq \frac{md}{nL} \mathrm{Var}(\hat{g}_{\star,m}) \geq \mathrm{Var}(\hat{g}_{\star,m}).$$

The last variance is cost-matched because m completions cost mL. □

J BRANCHING-LOCATION EXPERIMENTS

J.1 SAMPLING AND ESTIMATION

Each task uses 26 prompts, 64 replicates per prompt and temperature 1. HotpotQA uses Qwen2.5-3B after 60 search-training steps, `e5`/wiki-18 retrieval (top three passages), at most three searches, 500 generated tokens per turn, 500 retrieved tokens per search and 4,096 context tokens; candidates follow searches. MATH uses the base model after 20 GRPO steps, with candidates at $304/384/576/704$ tokens.

Each replicate has twelve independent rollouts; the first four form the skeleton, with three fresh continuations at every candidate. HotpotQA retains all 26 of 34 screened prompts showing skeleton rewards both ≥ 0.8 and < 0.8 (24 screening replicates; seeded samples from the first 1,000 training rows). MATH retains the first 20 and six eligible prompts from successive 48-candidate, 12-rollout screens, using pass rates in $(0.1, 0.9)$. Selection precedes collection.

The root-anchored estimator equation 4 uses one fixed $4{,}096$-dimensional signed-hash projection of full-parameter scores. Random placement is uniform; entropy placement maximises the rise in twenty-token mean entropy after a boundary. Without candidates, retain four independent completions. These retrospective rules are not previsible; variance ratios need not equal MSE ratios.

Ratios use sample trace variance. Match mean sampled-token cost, counting shared prefixes once, by interpolating between neighbouring independent-completion counts and their variances. Paired bootstrap intervals use 2,000 replicate resamples with re-matched costs; prompt-mean intervals use 10,000 prompt resamples.

J.2 PLACEMENT AND CALIBRATION

For theory-guided placement, siblings 2–3 provide plug-in V, G, C, B estimates for equation 55; evaluate the minimising candidate with the original suffix and sibling 1. Probe cost is excluded, making this an oracle comparison. The sibling split does not test sensitivity to the fixed projection.

For Figure J.1, all four continuations supply the prediction's empirical law. For replicate b, the measured contribution is

$$y_b(f) = \frac{B}{B-1} \frac{\|\hat{g}_{\beta,\varnothing}(\mathcal{T}_{f,4})^{(b)} - \bar{g}_{f,4}\|^2 - \|\hat{g}_{\beta,\varnothing}(\mathcal{T}_4)^{(b)} - \bar{g}_4\|^2}{\widehat{\mathrm{Var}}(\hat{g}_{\beta,\varnothing}(\mathcal{T}_4))}, \qquad B = 64,$$

where $\mathcal{T}_4$ is the skeleton and $\mathcal{T}_{f,4}$ adds a four-child fork at f; bars are replicate means. Both axes use skeleton-variance normalisation; absent vertices contribute zero. Prediction and measurement reuse continuations: calibration is not independent validation.

J.3 PER-PROMPT RESULTS

Table J.1 gives two-child ratios for all 52 prompts.

K GRADIENT-NOISE PROFILE ESTIMATION

For groups of $n = 4$ completions, estimate equation 12 by the sample trace variance of K independently resampled group gradients (denominator $K - 1$), retaining terminated rollouts unchanged. Average within each prompt, then equally across prompts; intervals use 2,000 prompt-bootstrap resamples. Scores include only policy-generated tokens, without length or reward-standard-deviation normalisation. A fixed $4{,}096$-dimensional signed-hash sketch per prompt preserves addition; squared norms are unbiased over sketch randomness for fixed gradients.

MATH. The step-20 policy uses temperature 1, horizon $1{,}024$ and boxed-answer exact-match reward, with two groups per prompt and $K = 8$. Checkpoints $320/384/480/720$ come from length quantiles in a separate pilot (16 prompts, eight rollouts each). Prompts are the first 16 of 20 with pass rate in $(0.1, 0.9)$ from a 48-candidate, 12-rollout test-split screen.

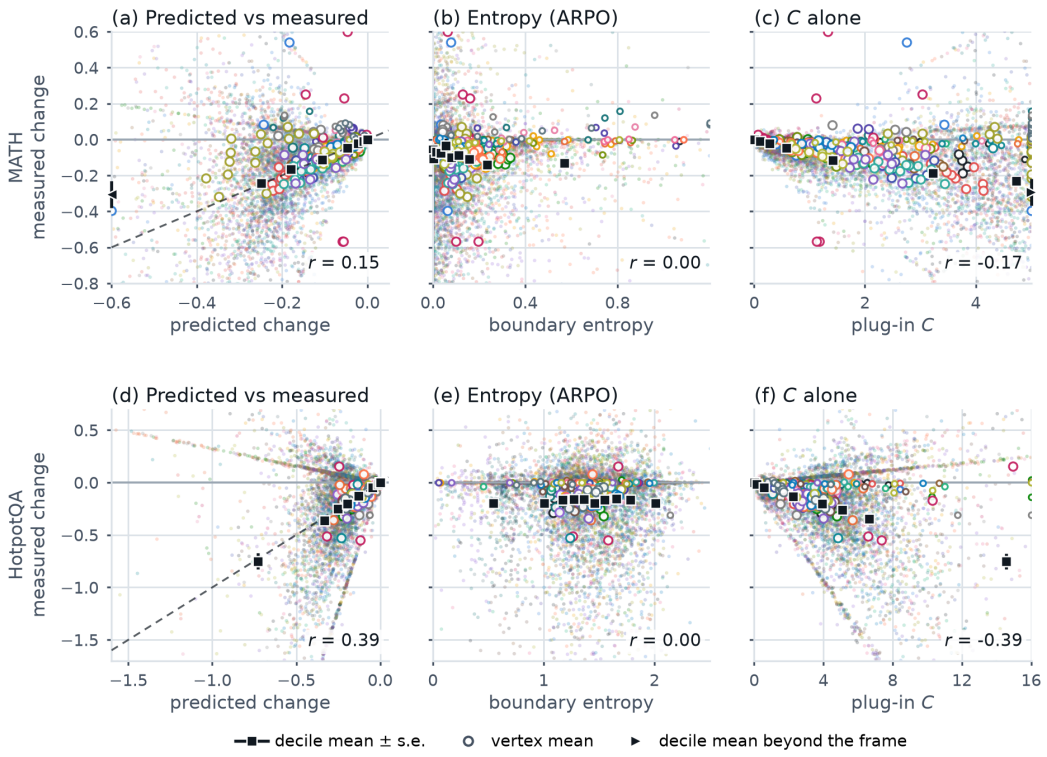

Figure J.1: Realised variance changes against plug-in predictions, boundary entropy and gradient noise C. Top: MATH (13,209 forks); bottom: HotpotQA (9,966), with 26 prompts each. Squares: decile means ± s.e.; triangles: means outside the frame; circles: candidate-vertex means.

HotpotQA. Use the first 16 eligible prompts from Appendix J.1, with 64 replicates each. After searches 1–3, the first four skeleton rollouts and their three fresh continuations give $K = 4$ group gradients; the root uses three disjoint groups from the twelve rollouts. The token axis is mean generated tokens through each search.

Prompt sensitivity. Remaining MATH variance at the four checkpoints is $0.70/0.67/0.49/0.19$ on 16 random prompts versus $0.62/0.47/0.23/0.08$ on selected prompts, with overlapping bootstrap intervals. These coarse estimates do not establish the theorem's pointwise monotonicity assumption.

L Training grid: setup and limitations

L.1 Training and evaluation

Task and optimisation. We train `Qwen2.5-3B base` on HotpotQA's 90,447 training questions in Search-R1, using a wiki-18 passage index with `e5-base-v2` retrieval and three passages per query. Rollouts allow at most three searches and 500 generated tokens per turn; retrieved observations are excluded from the loss. Rewards are 1 for a correct, well-formed answer, 0.2 for a well-formed incorrect answer, and 0.1 otherwise. All runs use dual-clip PPO with a token-mean loss, learning rate 10^{-6}, KL and entropy coefficients 0.001, and 180 training steps with eight optimiser updates per step. Following the respective released recipes, warm-up occupies 28.5% of branching runs and 50% of independent-completion runs.

Evaluation. We evaluate every final checkpoint with greedy decoding on 22,523 benchmark questions: HotpotQA (7,405), 2WikiMultiHopQA (12,576), MuSiQue (2,417), and Bamboogle (125). Scores are exact match without format credits, averaged equally across the four datasets. Table 2 reports every seed, including failed runs. Training curves use an 875-question subset evaluated

MATH

prompt	random	entropy	theory
4	0.85	0.91	**0.67**
6	1.13	1.86	**0.99**
13	1.00	1.04	**0.84**
15	1.02	0.97	**0.66**
17	0.98	1.01	**0.84**
18	0.93	0.96	**0.77**
19	1.12	1.20	**0.95**
20	0.97	1.03	**0.97**
23	1.12	1.08	**1.03**
24	0.99	0.95	**0.83**
26	1.02	0.99	**0.89**
27	0.99	0.94	**0.91**
33	1.08	0.96	**0.89**
36	1.32	1.27	**0.85**
37	1.00	1.00	**0.82**
40	1.07	1.09	**0.85**
41	0.87	0.81	**0.80**
42	0.99	0.98	**0.72**
43	0.90	0.92	**0.76**
46	0.96	0.99	**0.88**
51	1.01	1.00	**0.81**
52	1.03	1.02	**0.86**
53	1.04	0.98	**0.85**
56	1.01	0.97	**0.92**
59	1.03	1.02	**0.88**
60	1.05	1.03	**1.05**

HotpotQA

prompt	random	entropy	theory
0	0.97	0.98	**0.60**
3	0.91	0.87	**0.62**
5	0.98	1.00	**0.82**
1	1.03	1.02	**0.75**
82	1.01	1.02	**0.65**
84	0.96	0.98	**0.68**
2	0.78	0.75	**0.43**
441	0.95	0.98	**0.60**
515	0.91	0.92	**0.81**
546	0.97	0.97	**0.70**
615	0.91	0.98	**0.72**
699	0.90	0.79	**0.51**
66	1.00	0.95	**0.75**
125	1.14	0.68	**0.31**
184	0.68	0.81	**0.38**
276	0.95	0.60	**0.55**
214	1.00	0.93	**0.73**
314	0.99	0.97	**0.75**
342	1.01	0.98	**0.73**
364	0.97	0.77	**0.54**
100	0.96	0.94	**0.79**
186	1.03	0.90	**0.60**
201	0.93	0.99	**0.48**
244	0.96	0.98	**0.70**
55	1.04	0.90	**0.59**
410	0.98	0.96	**0.67**

Table J.1: Per-prompt variance ratios for a single fork with two children, relative to independent completions at matched cost (64 replicates per prompt). Theory uses held-out oracle inputs.

every 20 steps. An upstream validation bug makes the independent-completion runs sample at temperature 1, whereas branching runs use greedy decoding. This affects the curves and the failure labels derived from them; all final scores use the same greedy evaluation procedure.

Sampling and budget. Independent sampling uses 448 prompts with four completions each. Random branching uses 448 prompts with two three-leaf trees each (Tree-GRPO's `m2 n2 l1 k3`): each seed chain receives two expansions at uniformly sampled vertices. Entropy branching uses 168 prompts with a budget of sixteen rollouts each (ARPO), beginning with eight root chains. After a tool result, a chain forks with probability $\mathrm{clip}(0.5 + 0.2\,\Delta H, 0, 1)$, where ΔH is the entropy rise over the first twenty tokens; unused budget restarts chains at the root. Budgets are approximately matched in whole-trajectory equivalents, counting a shared prefix once: 1,792 for independent sampling, 1,792 nominally for random branching under the released accounting of two full chains plus four half-chain expansions per prompt, and about 1,800 measured for entropy branching. These are not exact action-budget matches; entropy runs also see $2.7\times$ fewer prompts per step. Realised costs appear in Table 2.

Scope of the unbiasedness guarantee. Neither released branching implementation satisfies Assumption 3.1. Tree-GRPO selects expansion vertices with replacement after completing each seed chain, so a vertex's child count depends on the realised continuation length. All three leaves are retained. ARPO's entropy trigger is previsible, but its shared rollout budget and root restarts make child counts depend on other chains' outcomes. Thus, applying $\widehat{g}_{\beta,\alpha}$ to these samplers does not inherit the theorem's unbiasedness guarantee. Admissible variants with expansion depths chosen in advance or separate per-chain fork budgets were not evaluated.

Estimators and scaling. The shipped GRPO and ARPO estimators standardise rewards within each prompt's group. Tree-GRPO sums standardised rewards over the prompt's six leaves and within each three-leaf tree. Our estimator $\widehat{g}_{\beta,\alpha}$ from Section 2 uses either the root as the only anchor or every

estimator	seed	Hotpot	2wiki	MuSiQue	Bamb	Avg.
GRPO (shipped)	1	40.5	39.9	17.0	38.4	34.0
	2	40.3	38.2	16.9	37.6	33.2
	3	0.6	0.2	0.0	0.0	$0.2^{\dagger}$
$\hat{g}_{\star,4}$	1	32.0	30.8	8.5	17.6	$22.2^{\dagger}$
	2	40.1	41.2	17.1	37.6	34.0
	3	0.0	0.0	0.0	0.0	$0.0^{\dagger}$
Tree-GRPO (shipped)	1	42.3	42.6	17.7	36.8	34.9
	2	42.2	40.1	18.4	39.2	35.0
	3	42.3	38.6	17.8	39.2	34.5
$\widehat{g}_{\beta_{\mathrm{rand}},\varnothing}$	1	39.5	39.6	15.3	34.4	32.2
	2	41.5	42.3	18.2	40.8	35.7
	3	34.2	30.8	16.1	34.4	28.9
$\widehat{g}_{\beta_{\mathrm{rand}},\mathcal{V}^{\mathrm{f}}}$	1	6.0	3.2	2.6	8.8	$5.1^{\dagger}$
	2	2.8	2.0	0.6	0.8	$1.6^{\dagger\ddagger}$
	3	21.9	27.6	3.9	13.6	$16.8^{\dagger}$
ARPO (shipped)	1	39.8	37.8	15.4	32.8	$31.5^{\ddagger}$
	2	21.0	26.4	5.5	16.0	$17.2^{\dagger}$
	3	20.6	25.8	4.4	16.8	$16.9^{\dagger\ddagger}$
$\widehat{g}_{\beta_{\mathrm{ent}},\varnothing}$	1	41.0	40.2	16.9	40.0	34.5
	2	38.1	32.9	16.2	33.6	30.2
	3	41.2	39.7	18.5	37.6	34.2
$\widehat{g}_{\beta_{\mathrm{ent}},\mathcal{V}^{\mathrm{f}}}$	1	41.7	39.3	17.5	38.4	34.2
	2	42.9	41.5	17.5	36.0	34.5
	3	39.6	36.9	16.2	35.2	$32.0^{\ddagger}$
published chain GRPO (Ji et al., 2026)		39.0	36.3	15.2	36.8	31.8
published Tree-GRPO (Ji et al., 2026)		42.4	43.7	17.8	43.2	36.8

Table J.2: Exact match (0–100) at step 180 with greedy decoding on the evaluation sets of HotpotQA (7,405 questions), 2WikiMultiHopQA (12,576), MuSiQue (2,417), and Bamboogle (125), for every seed in Table 2. Avg. is the unweighted mean across datasets. † denotes failure to train and ‡ gradient freeze. The last two rows are published single-run results for Qwen2.5-3B.

fork as an anchor, without standard-deviation normalisation. To control the change in advantage magnitude, a fixed multiplier is set from the ratio of shipped to $\widehat{g}_{\beta,\alpha}$ advantage token-RMS on the same step-1 batch, multiplied by a common calibration factor 0.664. The resulting scales are 7.1 for random branching, 35.3 and 34.6 for entropy branching with root and local anchors, and 1.39 for RLOO ($\hat{g}_{\star,4}$), which uses the $n/(n-1)$ leave-one-out correction.

Comparison limits. The grid compares released training recipes rather than isolating the sampling rule alone: warm-up, prompt counts, and advantage scales differ. Two unmarked root-anchored runs also underwent repeated preemptions and resumes, which reshuffled the data order; the entropy run included a restart from step 39. These differences limit causal attribution of accuracy and stability differences.

L.2 Failures and reporting

A seed is marked † (failed to train) if its final subset validation score is below 25 or below half its best score, and ‡ (gradient freeze) if persistent non-finite gradients cause all remaining updates to be skipped. These criteria apply to every seed. Eight runs fail to train and four freeze, with two runs meeting both criteria. Frozen checkpoints remain unchanged despite apparently stable reward and validation curves; the cause of the freezes is unresolved.

Table 2 reports every seed's final score. Healthy means and sampling summaries exclude all marked seeds and therefore describe performance conditional on completing training without either failure. With only three seeds per row, failure-rate comparisons are inconclusive: even three unmarked runs versus none gives a two-sided Fisher exact $p = 0.1$. We evaluate the final checkpoint rather than selecting each run's best; fifteen of the 24 runs attain a higher subset score earlier in training.

L.3 Realised cost and branching locations

Table 2 reports realised sampling costs. Whole-trajectory equivalents are nominal for independent and random sampling, and measured from rollout counters for entropy sampling. Failed policies can appear cheaper when they stop using the search tool, so cost differences alone do not establish sampling efficiency.

Short diagnostic runs place Tree-GRPO's expansions at agent steps 0–1. In one ARPO batch, 57 of 71 forks occur after one tool call, 11 after two, and 3 after three. For context, the separate search-task profile in Section 5.1 reveals 14% of gradient variance by the first tool turn and 93% by the third. These diagnostics describe branching locations; they do not establish their effect on training.

L.4 Comparison with published results

Using the released implementations, our first-seed scores are 34.9 for Tree-GRPO and 34.0 for independent GRPO, compared with the published 36.8 and 31.8 (Ji et al., 2026). Means over unmarked seeds are 34.8 and 33.6, respectively. This is not a matched replication: our base-model, batch-448 protocol differs from the released Instruct-model, batch-512 defaults, and the published results are single runs. We also corrected an unassigned learning-rate variable and added resume support. We do not attribute the difference in reported gains to any single factor.