TREEHCA: HINDSIGHT-GUIDED EXPLORATION AND CREDIT ASSIGNMENT FOR MULTI-TURN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome-supervised group-relative policy optimization (GRPO) gives every token in an LLM agent's trajectory the same advantage, leaving useful and unhelpful decisions undifferentiated. Hindsight credit assignment (HCA) can refine this feedback, but its estimates depend on trajectories containing informative alternatives at consequential histories. We introduce TREEHCA, which couples *sampling via hindsight ratios* to generate these alternatives with *hindsight credit assignment* to evaluate their contribution. We use changes in reference-answer likelihood to revisit intermediate histories and select alternative continuations, forming a tree of reasoning and tool interactions. This exploration changes the distribution of sampled continuations. We derive the corresponding inverse-hindsight importance correction, including an exact finite-pool identity, and use its structure in a normalized tree backup that propagates terminal outcomes into turn-level advantages. This common hindsight structure is TREEHCA's central design principle, linking informative exploration to hindsight-guided credit assignment. Across seven question-answering benchmarks and three Qwen backbones, TREEHCA outperforms every baseline in aggregate exact match, validating our paired design for multi-turn LLM training.
Our code is available at GitHub: anonymous.4open.science/r/TreeHCA.

1 INTRODUCTION

An LLM agent solves a multi-turn task through reasoning steps and tool calls. With terminal verification, success feedback arrives only after a rollout is completed. Outcome-supervised group-relative policy optimization (GRPO) converts this feedback into a trajectory advantage and *assigns it to every generated token* (Shao et al., 2024). A successful trajectory can reinforce unhelpful turns, while a failed one can penalize useful decisions. Therefore, a central challenge in providing feedback for training multi-turn LLM agents is to score intermediate decisions by their contribution to success.

How should we assign credit to intermediate decisions? Hindsight credit assignment (HCA) relates an action's probability to the subsequent outcome (Harutyunyan et al., 2019). An action that appears

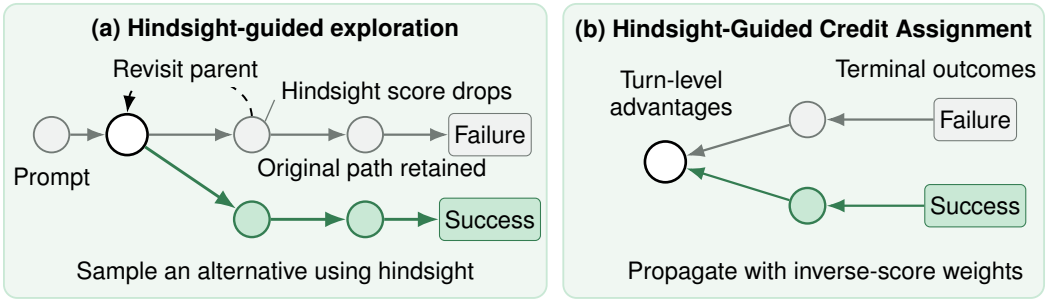

Figure 1: TREEHCA pairs exploration with credit assignment. (a) A drop in hindsight score identifies a parent history to revisit. The sampled alternative continues alongside the retained original path, here illustrated with successful and unsuccessful outcomes. **(b)** Outcome signals propagate backward using normalized inverse-score weights to construct turn-level advantages.

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.

disproportionately often in successful continuations receives a larger success-conditioned hindsight ratio. This ratio has an exact change-of-measure interpretation, which connects outcome-conditioned observations to the action value under the original policy. In a language agent, the space of interaction histories is large, so the usefulness of this construction depends on obtaining informative alternatives from comparable histories. GiGPO forms groups of comparable states to obtain step-level relative advantages, and HCAPO uses outcome-conditioned LLM scores to refine trajectory feedback (Feng et al., 2025; Tan et al., 2026). These methods make intermediate credit available from policy-generated chains. However, their estimates critically depend on these chains containing informative alternatives. In practice, such alternatives may be sparse at consequential histories, leaving the sampled trajectories insufficiently informative for turn-level credit assignment.

It is therefore crucially important to gather this information by revisiting histories from which alternative decisions can change the outcome. When an observed turn reduces the likelihood of success, we need to return to its parent and sample new continuations while retaining the original path. This produces comparisons under a shared prefix. Entropy-guided exploration allocates sampling effort to uncertain decisions (Dong et al., 2026; Zong et al., 2026). However, uncertainty alone does not reveal a decision's contribution to success. We instead use hindsight to guide exploration toward successful outcomes. Hindsight scores identify turns worth revisiting and promising alternatives to sample from their parent histories. We therefore introduce *hindsight-guided sampling* as an exploration strategy for obtaining the information needed to evaluate intermediate decisions.

We then make a complementary connection that is key to this paper: how we gather evidence through exploration determines how we must weight it when estimating turn-level advantages. By selecting continuations through hindsight, we change their distribution in the rollout tree. We must therefore account for that selection when estimating advantages for the original rollout policy. For a hindsight-tilted proposal, we derive the inverse hindsight ratio as the required importance weight. Our central insight is this correspondence: *the hindsight ratio guides exploration toward informative continuations, and its inverse accounts for that sampling preference when estimating credit.*

We develop this correspondence in **TREEHCA**. Its core idea has two parts. First, we use *hindsight-guided sampling* to generate informative trajectories. Changes in the model's likelihood of the reference answer identify turns to revisit and guide our selection of alternative continuations. Second, we use *credit assignment to samples* to accord advantages to intermediate turns according to the outcomes of their continuations. We propagate group-normalized terminal rewards through the tree using inverse hindsight weights to counterbalance the selection preference. The resulting turn-level signals reflect the successful and unsuccessful continuations supported by each intermediate decision. The key to TREEHCA is that these operations follow the same hindsight change of measure: the score that directs exploration also determines the inverse correction used to compute credit. This pairing is the central design principle behind the gains evaluated in Section 6. See Figure 1.

We make three contributions. **(1) Hindsight-guided exploration.** We formulate an exploration rule that uses turn-level hindsight scores to select histories and alternative continuations, connecting the exact hindsight ratio to candidate action values. **(2) Hindsight-guided credit assignment.** We derive the importance correction for hindsight-guided selection, including an exact finite-pool identity, and use its inverse-score structure to construct turn-level advantages through a normalized tree backup. **(3) Experimental validation of the paired construction.** We evaluate TREEHCA on seven question-answering benchmarks with Qwen2.5-3B-Instruct, Qwen3-4B, and Qwen3-8B. Across the reported model and task settings, TREEHCA outperforms all compared methods in aggregate exact match (EM). The paired design also achieves the strongest EM and F1 in the component study.

2 RELATED WORK

Intermediate credit from terminal outcomes. HCA expresses action values through outcome-conditioned action probabilities (Harutyunyan et al., 2019). For LLM agents, GiGPO constructs groups around recurring or comparable environment states and combines step-level and trajectory-level advantages (Feng et al., 2025). HCAPO uses the LLM as a hindsight critic, normalizes its outcome-conditioned action scores within a trajectory, and combines them with group-relative feedback (Tan et al., 2026). IGPO uses changes in reference-answer likelihood as information-gain rewards (Wang et al., 2026). These mechanisms extract intermediate learning signals from linear rollouts. TREEHCA uses outcome relevance to select the continuations supplying these signals.

Tree sampling and turn-level learning. Tree-GRPO shares prefixes in LLM-agent rollouts and constructs relative advantages within and across trees (Ji et al., 2026). In diffusion models, TreeGRPO backs terminal advantages through a denoising tree (Ding & Ye, 2026). AEPO allocates sampling using entropy-based information proxies and shapes the optimization signal, while AT^{2}PO combines entropy-guided exploration with an entropy-weighted tree backup (Dong et al., 2026; Zong et al., 2026). IGRPO uses reference-answer likelihood to guide adaptive rollout expansion (Zhang et al., 2026). These approaches motivate examining exploration and credit together. Our focus is the sampling distribution induced by hindsight selection and the corresponding inverse weighting in the backup. The separate policy-update ratio compares the updated and behavior policies. Appendix A provides a detailed review of these sampling and credit mechanisms.

3 Problem Formulation

We consider an agent that starts from a question x and alternates reasoning, tool use, and observation before producing an answer. At turn t, its history $h_{<t}$ contains the prompt and preceding interactions. The policy generates a reasoning sequence s_t and an action a_t, which we treat jointly as a decision $u_t = (s_t, a_t)$. Executing a tool action produces an observation o_t. A completed turn is $n_t = (u_t, o_t)$, and a trajectory is $\tau = (x, n_0, \ldots, n_T)$. A final-answer action terminates the trajectory. The verifier supplies a binary terminal reward $R(\tau) \in \{0, 1\}$, and the objective is

$$J(\theta) = \mathbb{E}_{\tau \sim P_{\pi_\theta}}[R(\tau)], \tag{1}$$

where P_{π_θ} includes both policy decisions and environment responses.

During one rollout-collection round, we hold the behavior policy $\pi = \pi_{\theta_{\text{old}}}$ fixed. All success probabilities and hindsight distributions below refer to continuations under this policy. Write $V^\pi(h) = \mathbb{P}_\pi(R = 1 \mid h)$ and $Q^\pi(h, u) = \mathbb{P}_\pi(R = 1 \mid h, U = u)$. The latter averages over the observation following u and subsequent policy decisions. For a completed child $c = (u, o)$ of n,

$$p_\pi(c \mid n) = \pi(u \mid h(n)) P_{\text{env}}(o \mid h(n), u). \tag{2}$$

The history $h(n)$ includes the full prefix ending at n. Multiple trajectories with shared prefixes form a tree $\mathcal{T}$. Its sibling nodes supply alternative completed turns from a common parent, and its leaves carry the verifier outcomes used for learning. We use the reference answer $y^\star$ during training to score these alternatives, and the learned agent uses its ordinary policy and search tool during evaluation.

4 Hindsight-Guided Exploration

We first develop TreeHCA's hindsight exploration policy. This is the sampling part of the paired construction: we use scores indicative of final outcome likelihood to decide where to gather informative alternatives. A turn that reduces the likelihood of success identifies a parent history worth revisiting. We sample from the history before that decision, so an alternative can change both the next action and the evidence subsequently available to the agent. Retaining the original continuation provides a comparison under the same prefix. We formalize this criterion through the exact hindsight distribution, then construct its practical score and tree-sampling rule. The terminal verifier evaluates each retained path, supplying the outcomes whose credit we compute in Section 5.

4.1 Sampling via Credit Assignment

For a supported decision u at history h and an outcome r of positive probability, the hindsight ratio is

$$\rho_{\text{HCA}}(h, u; r) = \frac{\mathbb{P}_\pi(U = u \mid h, R = r)}{\pi(u \mid h)} = \frac{\mathbb{P}_\pi(R = r \mid h, U = u)}{\mathbb{P}_\pi(R = r \mid h)}. \tag{3}$$

Bayes' rule gives the second equality. A ratio above one means that u is more prevalent among continuations with outcome r than under the rollout policy. For success,

$$\rho_{\text{HCA}}(h, u; 1) = \frac{Q^\pi(h, u)}{V^\pi(h)}, \qquad \log \rho_{\text{HCA}}(h, u; 1) = \log Q^\pi(h, u) - \log V^\pi(h). \tag{4}$$

Thus a negative log ratio identifies a decision after which success is less likely than it was at the parent history. Exploring from that parent supplies alternatives to the observed decision.

The same quantity ranks the alternatives. The following identity instantiates the return-conditioned HCA result of Harutyunyan et al. (2019) for our turn-level process.

Lemma 1 (Hindsight ranking of candidate decisions)**.** *For $\pi(u \mid h) > 0$, let $T(h, \pi)$ be the distribution of complete continuations from h under π and the environment. Then*

$$Q^\pi(h, u) = \mathbb{E}_{\tau \sim T(h,\pi)} \left[R(\tau) \rho_{\mathrm{HCA}}(h, u; R(\tau)) \right]. \tag{5}$$

For binary terminal rewards and $V^\pi(h) > 0$, this reduces to $Q^\pi(h, u) = V^\pi(h) \rho_{\mathrm{HCA}}(h, u; 1)$.

The parent value is common to all candidates, so the exact hindsight ratio ranks them by expected terminal reward. A low ratio identifies an observed turn to revisit, while a high ratio identifies an attractive alternative from its parent (appendix D.1).

4.2 Scoring Histories with the Policy LLM

We obtain a training-time proxy using the reference answer $y^\star = (y_1^\star, \ldots, y_J^\star)$. Following the use of answer likelihood as an information signal (Wang et al., 2026; Zhang et al., 2026), we append a fixed answer-eliciting instruction $s^\star$ to a history and teacher-force its answer tokens. The score is the mean token log probability

$$\ell_\pi(h; y^\star) = \frac{1}{J} \sum_{j=1}^{J} \log \pi(y_j^\star \mid h, s^\star, y_{<j}^\star). \tag{6}$$

For a completed turn $c = (u, o)$, let $h^-(c)$ and $h^+(c)$ be the histories before and after that turn.

$$\delta(c) = \log \widehat{\rho}_c = \ell_\pi(h^+(c); y^\star) - \ell_\pi(h^-(c); y^\star). \tag{7}$$

This score measures how the realized turn changes compatibility with the reference answer. It includes the information supplied by the tool observation. A negative score identifies a reduction in that compatibility and motivates collecting alternatives from $h^-(c)$. The teacher-forced likelihood provides a computable surrogate for the conditional success probabilities in Equation (4).

4.3 Selecting and Completing Alternatives

Starting from M complete policy rollouts, we identify turns with $\delta(c) < \kappa_{\mathrm{br}}$ for a fixed negative threshold. In each expansion round, we select up to K lowest-scoring turns. If fewer than K satisfy the threshold, uniform tie-breaking fills the allocation from the available turns. Each selected turn identifies its parent as a starting point for exploration, as illustrated in Figure 2.

At a selected parent, we draw N candidate decisions from π, execute their actions to obtain completed candidates $C_1, \ldots, C_N$, and score them. We sample one candidate index with

$$\alpha_i = \mathbb{P}(I = i \mid C_{1:N}) = \frac{\exp \delta(C_i)}{\sum_{j=1}^{N} \exp \delta(C_j)} = \frac{\widehat{\rho}_{C_i}}{\sum_{j=1}^{N} \widehat{\rho}_{C_j}}. \tag{8}$$

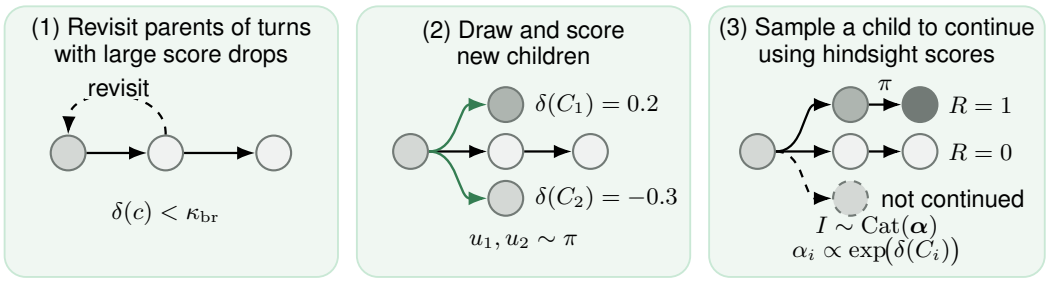

Figure 2: Hindsight-guided exploration. **(1)** A low score on an observed completed turn identifies its parent for resampling. **(2)** From that starting point, $N = 2$ policy decisions are executed and scored as completed candidates. **(3)** A categorical draw selects one using normalized hindsight scores. It continues under π to a verifier reward, while the original child and subtree remain. Scores and outcomes are illustrative.

Algorithm 1 Hindsight-guided tree construction

Require: Prompt x, reference answer $y^\star$, frozen policy π, initial count M, rounds L, exploration budget K, candidates N, threshold $\kappa_{\mathrm{br}} < 0$.

```
Initialize T with M complete policy rollouts from x.
for ℓ = 1, ..., L do
    Score each non-root node by b(c) = min{δ(c), κ_br}.
    Select up to K nodes with the smallest scores, breaking ties uniformly.
    for each selected node c do
        Draw N decisions from π(· | h⁻(c)) and execute them.
        Score the completed candidates and sample I using Equation (8).
        Attach C_I to pa(c) and complete it under π, retaining c and its subtree.
return T with its terminal verifier rewards.
```

Only C_I is continued to termination, and the original child and its subtree remain available for learning. This spends N turn executions and one full continuation per expansion. Stochastic selection allocates the continuation budget across scored alternatives.

Algorithm 1 gives the complete construction. The tree now contains both original rollouts and selectively retained alternatives. Their outcomes provide comparisons under a shared prefix, while their selection probabilities determine how these comparisons should contribute to a rollout-policy expectation. This is the estimation task addressed next.

5 Hindsight-Guided Credit Assignment

We now develop the complementary part of the pair which includes assigning credit to the samples gathered by TreeHCA's exploration policy. Hindsight-guided selection favors high-scoring continuations, while the learning signal must reflect their value under the original rollout policy. The same scores that guide exploration therefore determine the inverse weighting needed for estimation. We derive this correspondence for the specified sampling laws and use its inverse-score structure to construct turn-level advantages on the tree.

5.1 The Sampling and Estimation Correspondence

Fix a parent n and write $p(c) = p_\pi(c \mid n)$ for its completed-child kernel. For a positive score $s(c)$ with $Z = \mathbb{E}_p[s(C)] \in (0, \infty)$, consider the tilted proposal

$$q_s(c) = \frac{p(c)s(c)}{Z}, \qquad \frac{p(c)}{q_s(c)} = \frac{Z}{s(c)}. \tag{9}$$

The score specifies which children are favored in sampling, and its inverse supplies the change of measure back to the rollout policy. If $s(c)$ is the exact success-conditioned hindsight ratio of the completed child, this proposal is the success-conditioned child distribution on its support.

Proposition 1 (Credit-guided sampling and inverse correction)**.** *Let f be integrable under p. If q_s covers the support of pf, then*

$$\mathbb{E}_{q_s}\left[\frac{Z}{s(C)} f(C)\right] = \mathbb{E}_p[f(C)]. \tag{10}$$

For a finite pool $C_1, \ldots, C_N \overset{\text{iid}}{\sim} p$ and selection probabilities $\alpha_i = s(C_i)/\sum_j s(C_j)$, the corresponding identity is

$$\mathbb{E}\left[\frac{f(C_I)}{N\alpha_I} \,\middle|\, C_{1:N}\right] = \frac{1}{N}\sum_{i=1}^{N} f(C_i). \tag{11}$$

Consequently, averaging over the pool recovers $\mathbb{E}_p[f(C)]$.

The finite pool also determines the selected child's marginal law. For N candidates and a fixed first candidate c, symmetry gives

$$q_N(c) = Np(c)\,\mathbb{E}\left[\frac{s(c)}{s(c) + \sum_{j=2}^{N} s(C_j)}\right], \tag{12}$$

where the expectation is over the remaining independent policy candidates. At $N = 1$, selection follows p. For scores bounded above and away from zero, concentration of the remaining scores around their mean gives $q_N(c) \to p(c)s(c)/Z$. Thus the direct tilt describes large-pool selection, while Equation (11) provides an exact correction at every finite pool size.

Appendix D.2 proves these identities by a change of measure and a conditional weighted sum, and gives a two-child illustration. For the finite-pool selection in Equation (8), the weight is the mean candidate score divided by the selected score. The identity holds for any strictly positive score. It also holds for a conditionally unbiased estimate of $f(C_I)$ after averaging over continuation randomness.

The direction of this weighting follows the target distribution. In Lemma 1, the hindsight ratio multiplies a return sampled under the rollout policy to recover an action-conditioned value. In Proposition 1, the sampling law itself favors children with large scores, so recovering a rollout-policy expectation uses the inverse score. For the exact hindsight score, these identities express the two directions of the conditional-distribution change of measure.

These identities explain why exploration and credit must be designed together. Hindsight-guided selection makes high-scoring continuations more common, so their observed frequency alone cannot represent the rollout policy. Inverse weighting compensates for this preference when computing a policy-based expectation. The direct-tilt identity gives the correction for score-proportional sampling, and the finite-pool identity gives it conditional on the sampled candidates. TREEHCA carries the same inverse-score structure into a normalized backup over retained children, propagating their outcome signals to intermediate turns. Appendices D.3 and D.4 provide the derivations.

5.2 FROM LEAF REWARDS TO TURN-LEVEL ADVANTAGES

For the leaves belonging to one prompt, let μ_R and σ_R be the mean and standard deviation of the verifier rewards. Initialize a leaf ℓ with its group-relative signal

$$\widehat{A}^{\mathrm{tree}}(\ell) = \begin{cases} \big(R(\ell) - \mu_R\big)/\sigma_R, & \sigma_R > 0, \\ 0, & \sigma_R = 0. \end{cases} \tag{13}$$

The zero-variance case supplies a zero learning signal when every trajectory has the same outcome. The tree then propagates these signals toward the root. For the observed children $\mathrm{Ch}(n)$, define

$$\widetilde{w}_{n,c} = \frac{\widehat{\rho}_c^{-1}}{\sum_{c' \in \mathrm{Ch}(n)} \widehat{\rho}_{c'}^{-1}}, \qquad \widehat{A}^{\mathrm{tree}}(n) = \sum_{c \in \mathrm{Ch}(n)} \widetilde{w}_{n,c} \widehat{A}^{\mathrm{tree}}(c). \tag{14}$$

The recursion proceeds in reverse topological order. A node with one child inherits that child's signal. At a parent with multiple children, the inverse scores counterbalance the preference toward high-scoring alternatives, and normalization produces a convex combination of the child signals.

Under independent sampling from the specified tilted proposal, self-normalized inverse-score weighting consistently estimates the base-policy expectation. The tree operator applies this normalization to a finite, adaptively constructed tree with proxy scores. Its finite-tree behavior is evaluated through the credit-rule ablation in Section 6.3. The exact identities in Proposition 1 explain the correction's form, while the normalization preserves the scale and common shifts of the group-relative signal. The appendix states the estimator conditions and separates finite-pool selection from the retained siblings.

5.3 TURN-LEVEL POLICY OPTIMIZATION

Each policy-generated token receives the advantage of its completed turn. If token t of trajectory y_i belongs to node $n_{i,\kappa_i(t)}$, set $A_{i,t} = \widehat{A}^{\mathrm{tree}}(n_{i,\kappa_i(t)})$. Tool outputs and prompt tokens remain in the conditioning history and are masked from the loss. Let $M_{i,t}$ be the policy-token mask and $N_i = \sum_t M_{i,t}$. We use the GSPO-token surrogate (Zheng et al., 2025a), whose detached sequence ratio scales the token gradient:

$$r_i(\theta) = \exp\left[\frac{1}{N_i} \sum_t M_{i,t} \log \frac{\pi_\theta(y_{i,t} \mid h_{i,t})}{\pi(y_{i,t} \mid h_{i,t})}\right], \tag{15}$$

$$s_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid h_{i,t})}{\mathrm{sg}[\pi_\theta(y_{i,t} \mid h_{i,t})]} \, \mathrm{sg}[r_i(\theta)]. \tag{16}$$

Table 1: Qwen2.5-3B-Instruct comparison for chain-based baselines: EM (%). Darker to lighter green marks the three largest distinct values per column, with ties sharing a shade and maxima in bold. Underlined entries indicate methods (re-)evaluated in our setup. Each of these methods uses five trajectories.

Method	Type	Multi-hop QA				Single-hop QA			Avg.
		Hotpot	2Wiki	MuSiQue	Bamb.	NQ	TriviaQA	PopQA	
Backbone: Qwen2.5-3B									
Search-R1	Chain-based	32.4	31.9	10.3	26.4	34.1	54.5	37.8	37.4
ZeroSearch	Chain-based	27.4	30.0	9.8	11.1	41.4	57.4	44.8	39.1
GiGPO	Chain-based	36.8	36.2	12.2	29.6	43.1	60.5	46.1	43.3
IGPO	Chain-based	37.5	36.4	11.8	25.6	44.9	**61.8**	**46.5**	43.6
HCAPO	Chain-based	38.6	36.3	**14.8**	**64.5**	44.4	60.5	45.5	44.3
TREEHCA (n=5)	Tree-based	**39.9**	**39.4**	**14.8**	32.4	**45.3**	61.5	44.4	**44.9**

Here sg denotes stop-gradient. This policy-update ratio compares the updated and behavior policies. The inverse-hindsight weights in Equation (14) instead counterbalance continuation selection.

Let $\mathcal{B}_\pi(\cdot \mid x)$ denote the tree-generation law of Algorithm 1, and let $y_1, \ldots, y_G$ be its root-to-leaf trajectories. We optimize

$$\mathcal{J}_{\text{TREEHCA}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\mathcal{T}\sim\mathcal{B}_\pi(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{N_i} \sum_{t} M_{i,t} \min\{s_{i,t}(\theta)A_{i,t}, \text{clip}(s_{i,t}(\theta), 1-\varepsilon_l, 1+\varepsilon_r)\, A_{i,t}\} \right]. \tag{17}$$

Advantages are held fixed during the update. A shared-prefix turn receives the same node advantage in every descendant trajectory, with the sequence ratio evaluated for each complete trajectory.

6 EXPERIMENTS

We now conduct a set of experiments to answer two fundamental questions. *Does hindsight-guided exploration collect informative trajectories?* Chain-method comparisons place intermediate credit signals in context, and tree-method comparisons examine the choice of exploration rule. *Does inverse-hindsight weighting provide useful turn-level credit?* The component comparison changes exploration and credit rules in turn, thereby examining their contribution to the complete method.

6.1 SETUP AND COMPARISON PROTOCOLS

We evaluate search-based question answering with Qwen2.5-3B-Instruct against chain and tree baselines (Tables 1 and 2), and with Qwen3-4B and Qwen3-8B (Table 3) (Yang et al., 2025). Multi-hop benchmarks comprise HotpotQA, 2WikiMultihopQA, MuSiQue, and Bamboogle (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Press et al., 2023). Single-hop benchmarks comprise Natural Questions (NQ), TriviaQA, and PopQA (Kwiatkowski et al., 2019; Joshi et al., 2017; Mallen et al., 2023). The agent interleaves reasoning with queries to a Wikipedia index, receiving the top three passages per query. Qwen3 allows six tool calls and 6192 response tokens. Appendix table 7 gives the Qwen2.5 settings. Exact match (EM) is the percentage of questions whose normalized extracted answer matches a reference answer. Aggregate scores weight datasets by test-set size.

For Qwen3, we train on HotpotQA for 250 steps for multi-hop evaluation and NQ for 200 steps for single-hop evaluation. The tree configuration uses $M = 10$ initial rollouts, $K = 6$ exploration parents, and $L = 2$ expansion rounds. We train AT^{2}PO from its public implementation and apply TreeGRPO's likelihood-weighted backup (Ding & Ye, 2026) to search trajectories. These methods and TREEHCA share an expansion budget (Table 3). Published ReAct (Yao et al., 2023), GRPO, GSPO, and AEPO results complete the comparison. Appendix B provides further setup details.

6.2 HINDSIGHT-GUIDED EXPLORATION

Table 1 compares chain-based methods with TREEHCA. Search-R1 and ZeroSearch learn from completed reasoning-and-search trajectories (Jin et al., 2025; Sun et al., 2025). GiGPO constructs

Table 2: Qwen2.5-3B-Instruct comparison for tree-based baselines: EM (%). Colors follow Table 1.

Method	Type	Multi-hop QA				Single-hop QA			Avg.
		Hotpot	2Wiki	MuSiQue	Bamb.	NQ	TriviaQA	PopQA	
Backbone: Qwen2.5-3B									
Tree-GRPO	Tree-based	31.1	31.7	11.9	28.0	44.5	60.4	43.7	40.8
IGRPO	Tree-based	38.0	39.0	14.3	29.6	45.1	60.0	**45.2**	43.9
TREEHCA (n=5)	Tree-based	**39.9**	**39.4**	**14.8**	**32.4**	**45.3**	**61.5**	44.4	**44.9**

Table 3: Qwen3 comparison: EM (%). ReAct, GRPO, GSPO, and AEPO results are from Zong et al. (2026). TreeGRPO, AT^{2}PO, and TREEHCA use our shared training protocol. W. Avg. weights datasets by test-set size. Colors rank values within each backbone. Underlined entries indicate methods (re-)evaluated in our setup.

Method	Multi-hop QA					Single-hop QA			
	Hotpot	2Wiki	MuSiQue	Bamb.	W. Avg.	NQ	TriviaQA	PopQA	W. Avg.
Backbone: Qwen3-4B									
ReAct	30.42	32.92	12.83	44.80	30.01	26.75	53.53	35.34	41.31
GRPO	44.76	51.40	21.60	50.40	46.02	45.98	65.17	49.18	54.97
GSPO	47.07	49.25	22.68	50.40	45.69	46.01	64.24	48.50	54.28
TreeGRPO	46.40	51.10	20.27	53.60	46.26	46.31	64.91	49.75	55.19
AEPO	46.36	51.78	23.47	50.40	46.95	45.71	64.66	50.13	55.20
AT^{2}PO	47.44	52.24	23.71	**55.20**	47.62	48.34	64.94	49.92	55.55
TREEHCA	**49.23**	**52.73**	**24.15**	53.60	**48.52**	**48.78**	**66.61**	**50.19**	**56.38**
Backbone: Qwen3-8B									
ReAct	20.66	19.05	9.56	37.60	18.66	21.16	41.81	27.37	32.19
GRPO	47.01	53.69	21.35	54.40	48.03	45.70	67.42	50.17	56.29
GSPO	49.59	52.55	24.35	54.40	48.56	45.56	67.75	49.66	56.15
TreeGRPO	47.44	52.55	21.35	50.40	47.51	47.36	67.49	50.48	56.68
AEPO	49.17	52.97	24.01	54.40	48.62	49.92	68.31	51.77	57.94
AT^{2}PO	48.82	52.96	23.13	55.20	48.41	49.53	66.54	50.81	56.75
TREEHCA	**50.51**	**54.99**	**24.50**	**56.00**	**50.25**	**51.22**	**68.49**	**52.93**	**58.75**

turn-level comparisons by grouping comparable states, IGPO scores changes in reference-answer likelihood, and HCAPO weights actions using hindsight context (Feng et al., 2025; Wang et al., 2026; Tan et al., 2026). Table 2 adds tree-based baselines. Tree-GRPO uses shared-prefix comparisons (Ji et al., 2026), with its reported scores taken from the reproduction in Zhang et al. (2026). TREEHCA connects such feedback to the allocation of new continuations, explicitly revisiting histories before low-scoring decisions. It achieves the highest aggregate EM in this comparison, 44.9, exceeding HCAPO by 0.6 percentage points.

Table 3 examines tree construction and credit on the Qwen3 backbones. Relative to AT^{2}PO, TREEHCA increases the multi-hop weighted EM from 47.62 to 48.52 at 4B and from 48.41 to 50.25 at 8B. The corresponding single-hop increases are 55.55 to 56.38 and 56.75 to 58.75. Relative to TreeGRPO, the gains are 2.26 and 2.74 percentage points on multi-hop QA, and 1.19 and 2.07 points on single-hop QA. Across all seven methods, TREEHCA achieves the highest weighted EM for both model sizes and task families. At 8B, it exceeds the strongest published comparison, AEPO, by 1.63 points on multi-hop QA and 0.81 points on single-hop QA.

6.3 CREDIT ASSIGNMENT AND THE PAIRED CONSTRUCTION

We vary exploration and credit one at a time (Table 4). With the TREEHCA credit rule, hindsight exploration attains 51.76 EM, compared with 51.17 for entropy and 49.02 for random exploration.

Under hindsight exploration, TREEHCA credit reaches 51.76 EM and 65.88 F1, versus 49.60 and 62.74 with AT^{2}PO credit. Under entropy exploration, the EM scores are 51.17 and 50.78, respectively. Switching to TREEHCA credit therefore gains 2.16 EM and 3.14 F1 points under hindsight exploration, and 0.39 EM and 0.32 F1 points under entropy exploration. Under AT^{2}PO

Table 4: Component comparison on HotpotQA with Qwen3-4B (seed 42). Each row names the exploration rule and the credit rule. EM and F1 validation scores are percentages. Colors follow Table 1.

Exploration	Credit	EM	F1
Hindsight	**TREEHCA**	**51.76**	**65.88**
Entropy	TREEHCA	51.17	64.17
Entropy	AT^{2}PO	50.78	63.85
Hindsight	AT^{2}PO	49.60	62.74
Random	TREEHCA	49.02	61.60
Policy chains	GSPO	47.46	60.76

credit, however, entropy outperforms hindsight (50.78 versus 49.60 EM). The complete pair is strongest, and both metrics show that the effect of changing exploration depends on the credit rule.

6.4 VARIABILITY AND TRAINING COST

The accuracy comparisons raise two further questions: *How much does performance vary across training runs? What training cost accompanies the gains?* We examine the variation across three seeds and the wall-clock time required by the paired construction.

Variation across training runs. Table 5 reports the multi-hop results across three training seeds. The weighted EM is 48.52 ± 0.75 for Qwen3-4B and 50.25 ± 1.27 for Qwen3-8B. Across individual benchmarks, standard deviations range from 0 to 2.30 points at 4B and from 0.36 to 4.53 points at 8B. The aggregate gains over the strongest baseline are 0.90 and 1.63 points, respectively. Both gains exceed one standard deviation of TREEHCA's observed run variation, supporting reasonably consistent aggregate performance across these seeds. The larger deviations on some individual benchmarks indicate greater sensitivity to the training seed.

Table 5: TREEHCA multi-hop performance as mean $\pm$ one standard deviation over three training seeds. EM and its standard deviation are in percentage points. Means match Table 3. W. Avg. weights datasets by size.

Backbone	Hotpot	2Wiki	MuSiQue	Bamb.	W. Avg.
Qwen3-4B	49.23 ± 2.30	52.73 ± 0.34	24.15 ± 1.68	53.60 ± 0.00	48.52 ± 0.75
Qwen3-8B	50.51 ± 1.03	54.99 ± 1.55	24.50 ± 0.36	56.00 ± 4.53	50.25 ± 1.27

Training cost. The wall-clock training times on eight H200 GPUs are 34.40 versus 31.50 hours at 4B and 36.20 versus 32.58 hours at 8B for TREEHCA and AT^{2}PO, respectively. These correspond to increases of 9.2% and 11.1%. Under the shared expansion budget, the paired design therefore improves accuracy with 9–11% more training time. The timing comparison quantifies the computational cost accompanying the stronger credit signals.

7 CONCLUSION

TREEHCA connects exploration and intermediate credit through a common hindsight change of measure. Hindsight scores guide the selection of alternative continuations, and inverse weights account for that sampling preference when estimating advantages. We derive this relationship for a hindsight-tilted proposal and finite-pool candidate selection, then realize it through a normalized tree backup. Across the evaluated question-answering settings, the paired construction improves on the compared baselines, and component ablations connect its performance to the combination of exploration and credit. Together, the sampling identities and component experiments identify the pairing as the central design principle behind these gains.

AI USE STATEMENT

Claude Code and OpenAI Codex assisted with coding. Codex also assisted with literature organization, technical writing, results formatting, and LaTeX layout. We manually verified that the code correctly implemented the algorithms, then ran it on a batch processing system and verified the results. We take responsibility for the manuscript and artifacts.

REPRODUCIBILITY STATEMENT

Sections 4 and 5 specify the exploration rule and learning objective. Proofs and estimator conditions appear in appendix D, and configurations, result provenance, and uncertainty reporting in appendix B and section 6.4. Code: https://anonymous.4open.science/r/TreeHCA-83E6.

REFERENCES

Zheng Ding and Weirui Ye. TreeGRPO: Tree-Advantage GRPO for Online RL Post-Training of Diffusion Models. In *Proceedings of ICLR*, 2026.

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. Toward Generalized Web Agent Training: A Deep Dive into Entropy-Balanced Reinforcement Learning. In *Proceedings of the ACM Web Conference 2026*, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight credit assignment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/195f15384c2a79cedf293e4a847ce85c-Paper.pdf.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree Search for LLM Agent Reinforcement Learning. In *Proceedings of ICLR*, 2026.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. In *2nd Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

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. arXiv preprint arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the 20th European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297, 2025.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. ZeroSearch: Incentivize the Search Capability of LLMs without Searching, 2025. arXiv preprint arXiv:2505.04588.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight Credit Assignment for Long-Horizon LLM Agents, 2026. arXiv preprint arXiv:2603.08754.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn Search Agents. In *Proceedings of ICLR*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of ICLR*, 2023.

Chi Zhang, Guangming Sheng, Siyao Liu, Jiahao Li, Ziyuan Feng, Zherui Liu, Xin Liu, Xiaoying Jia, Yanghua Peng, Haibin Lin, and Chuan Wu. A Framework for Training Large Language Models for Code Generation via Proximal Policy Optimization. In *NL2Code Workshop of ACM KDD*, 2024.

Yijun Zhang, Fan Xu, Jiaxin Ding, Yule Xie, Shiqing Gao, Xin Ding, Haoxiang Zhang, Luoyi Fu, and Xinbing Wang. Information Gain-based Rollout Policy Optimization: An Adaptive Tree-Structured Rollout Approach for Multi-Turn LLM Agents, 2026. arXiv preprint arXiv:2607.06223.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization, 2025a. arXiv preprint arXiv:2507.18071.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21805–21830, 2025b.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. AT^{2}PO: Agentic Turn-based Policy Optimization via Tree Search. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24120–24143, 2026.

A Detailed related work

Intermediate credit assignment depends on both the information available at a decision and the continuations used to assess it. We organize the relevant methods by how they collect alternatives and translate outcomes into learning signals.

A.1 Hindsight distributions and practical action credit

Hindsight credit assignment (HCA) measures an action's relevance through its probability conditioned on a later state or return (Harutyunyan et al., 2019). Bayes' rule relates this posterior-to-policy ratio to the action's effect on future outcomes. Under exact conditionals and appropriate support, the forward ratio recovers action-conditioned values from policy-mixture outcomes, while the inverse ratio recovers policy-mixture expectations from action-conditioned outcomes. For binary rewards, the success-conditioned ratio ranks actions by expected reward at a common parent. These identities connect outcome relevance to a change of measure.

GiGPO obtains practical credit by combining episode-relative returns with discounted-return comparisons among actions at recurring anchor states (Feng et al., 2025). It matches environment-state keys, using sequence similarity for search-QA histories. These comparisons can link actions at different times or histories sharing an anchor. HCAPO uses the actor as a generative hindsight critic (Tan et al., 2026). It scores recorded actions with successful-outcome context, applies length and temperature normalization, divides each score by its trajectory mean, and clips the multiplier before combining discounted-return and trajectory-relative feedback. Both obtain intermediate signals from collected trajectories. TreeHCA uses the exact hindsight conditional for its ranking interpretation and reference-answer scores to direct the collection of alternatives.

A.2 Outcome and process feedback along policy trajectories

Outcome-supervised GRPO assigns each response's standardized reward to its generated tokens, comparing responses to the same prompt (Shao et al., 2024). Its process-supervised formulation instead accumulates intermediate reward-model scores. Search-R1 learns interleaved reasoning and retrieval from terminal answer rewards, using PPO with a learned value model by default and GRPO as an alternative (Jin et al., 2025). Retrieved passages condition subsequent decisions, while losses apply to generated tokens. ZeroSearch trains with a retrieval simulator and a curriculum of increasingly noisy observations, combining terminal answer feedback with experience under changing retrieval conditions (Sun et al., 2025).

Process rewards supply information about intermediate search decisions. StepSearch rewards gains in gold-evidence coverage, penalizes repeated retrieval, and adds answer and query supervision (Zheng et al., 2025b). PPO propagates these rewards through value estimates and generalized advantage estimation. IGPO instead uses changes in the reference answer's length-normalized log likelihood as dense turn rewards, normalizing process and outcome feedback separately (Wang et al., 2026). These methods refine credit within collected chains. In TreeHCA, answer information also determines which histories and decisions receive additional rollout budget.

A.3 Tree structure, exploration criteria, and credit aggregation

Shared-prefix trees compare continuations from a common history while reusing earlier interaction. The LLM-agent Tree-GRPO of Ji et al. (2026) expands randomly selected nonleaf steps and combines reward-normalized advantages within and across trees. Shared-prefix occurrences accumulate contributions from descendant trajectories. The diffusion-model TreeGRPO of Ding & Ye (2026) samples alternative denoising paths within a stochastic window and propagates normalized terminal advantages through child weights proportional to behavior-policy likelihoods. These distinct methods connect tree structure to credit in agent interactions and latent denoising, respectively.

Entropy provides another exploration criterion. AT^{2}PO ranks histories using sampled-token log probabilities and a penalty for repeated parent expansion, then aggregates leaf signals through entropy-weighted backups and applies a turn-level policy objective (Zong et al., 2026). AEPO uses a pilot trajectory to allocate sampling between the root and intermediate histories according to root and tool-related entropy (Dong et al., 2026). It spreads exploration through a consecutive-expansion

penalty and combines entropy-shaped advantages with a clipping modification that preserves positive-advantage gradients beyond the upper threshold. TREEHCA connects history and candidate selection to reference-answer compatibility and the corresponding sampling correction.

A.4 THE SAMPLING DISTRIBUTION AND POLICY OPTIMIZATION

IGRPO directly connects answer information to rollout allocation (Zhang et al., 2026). It combines length-normalized answer likelihood with improvement over the parent and exponentially transforms this score into expansion probabilities. Under a fixed policy, equal horizon, and growing per-layer budgets, its analysis gives an exponentially tilted trajectory distribution with a KL-regularized teacher interpretation. Group-relative training uses samples from this teacher. Our estimation target is the original rollout-policy expectation at a selected parent, connecting continuation selection to the reciprocal weighting that recovers that target.

For an independently generated candidate pool with positive selection probabilities, Equation (11) gives an exact correction: conditional weighting recovers the pool mean, and averaging over candidate generation recovers the base-policy expectation. The practical tree backup in Equation (14) normalizes inverse proxy scores over retained siblings. Under independent draws from the specified direct tilt, the corresponding self-normalized estimator is consistent (Appendix D.4). The sampling law therefore supplies the inverse-score form used in credit aggregation.

Policy optimization addresses a separate distribution change between rollout and updated policies. GSPO uses a length-normalized sequence ratio, while GSPO-token allows token-specific advantages through a stop-gradient construction (Zheng et al., 2025a). In TREEHCA, this optimizer acts on turn-level advantages after tree credit has been computed. The exploration rule and credit operator share a common principle: credit determines which alternatives are collected, and their collection law informs the credit assigned to their outcomes.

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 HYPERPARAMETERS

Appendix table 6 groups the Qwen3 settings shared by AT^{2}PO and TREEHCA according to their role in training. The optimizer and learning rate govern parameter updates, while clipping controls the likelihood-ratio range in the policy objective. The training and PPO mini-batch sizes specify the granularity of these updates. Prompt and response lengths are measured in tokens, and the tool-call limit bounds the number of external interactions along a trajectory.

The rollout settings describe how trajectories are collected. The tree starts with M complete trajectories and adds alternatives from up to K selected parents in each of L expansion rounds. These controls specify the allocation of exploration across histories and rounds, complementing the length and interaction limits on each trajectory.

Appendix table 8 gives the additional controls for hindsight-guided exploration. Score smoothing stabilizes the comparison of reference-answer likelihoods when these likelihoods are small. The negative exploration threshold identifies completed turns that reduce the score, and the sibling penalty lowers the priority of parents that already have alternatives. The candidate count N determines how many completed turns are scored at a selected parent before choosing a continuation.

B.2 RUNTIME

All training experiments for QA tasks were conducted on a high-performance computing cluster equipped with 8× NVIDIA H200 GPUs. Appendix table 9 reports wall-clock training time for Qwen3-4B and Qwen3-8B under AT^{2}PO and TREEHCA.

B.3 CODE DETAILS

Our code is built upon the codebase for AT^{2}PO (Zong et al., 2026), which itself is built on the VeRL framework (Sheng et al., 2025; Zhang et al., 2024).

Table 6: Qwen3 training and rollout settings shared by AT^{2}PO and TREEHCA.

Hyperparameter	Value
Policy optimization	
Optimizer	AdamW
Learning rate	1e-6
Clip_ratio	0.2
Training batch size	64
PPO mini-batch size	8
Sequence and interaction limits	
Maximum prompt length	2000
Maximum response length	6192
Maximum tool calls	6
Rollout allocation	
Rollout_n	16
Initial rollout count M	10
Selected parents K	6
Expansion rounds L	2

Table 7: Rollout settings for Qwen2.5-3B experiments.

Hyperparameter	Value
Training	
Number of training steps	200
Exploration threshold κ_{br}	-0.14
Sequence and interaction limits	
Maximum turns	4
Maximum rollout length	512
Rollout allocation	
GRPO rollout count n	5
TREEHCA candidate children N	2

Table 8: TREEHCA score and exploration controls.

Hyperparameter	Value
Reference-answer score smoothing $\epsilon_{\mathrm{smooth}}$	0.05
Exploration threshold κ_{br}	-0.14
Sibling penalty γ	0.05
Candidates per selected parent N	2

Table 9: Wall-clock training time in hours on 8 H200 GPUs.

Model	AT^{2}PO	TREEHCA
Qwen3-4B	31.50	34.40
Qwen3-8B	32.58	36.20

C ADDITIONAL COMPONENT MEASUREMENTS

Table 4 reports EM and F1 for the component comparison. The corresponding validation scores recorded for the six rows are 51.56, 50.00, 49.41, 48.05, 47.66, and 39.26, respectively. These values preserve the distinction between the validation measurement and the EM and F1 metrics used in the main comparison. All six rows use Qwen3-4B on HotpotQA with seed 42.

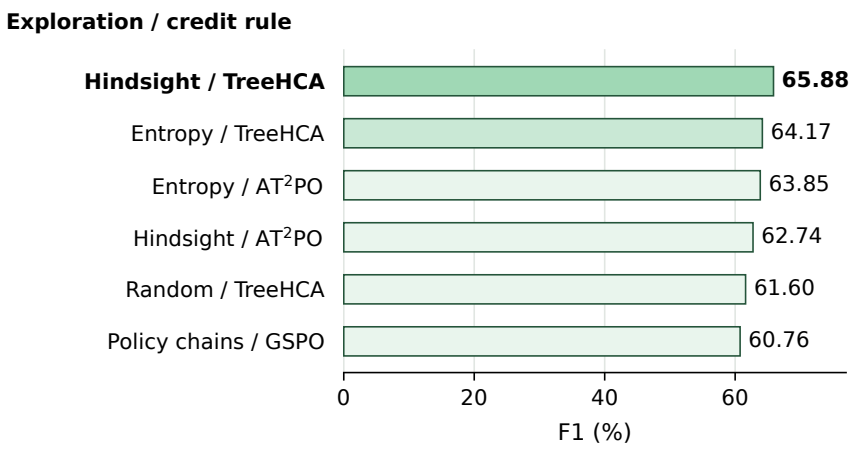

Figure 3: F1 for the component comparison in Table 4. The figure reports the F1 column, including 65.88 for the complete hindsight-guided exploration and credit pair.

D Hindsight Identities and Estimator Conditions

D.1 The Return-Conditioned HCA Identity

Lemma 1 instantiates the return-conditioned action-value identity of Harutyunyan et al. (2019), Theorem 5. The states of our process are complete interaction histories. A composite turn $u = (s, a)$ is an action, the environment supplies its observation, and the undiscounted return equals the terminal verifier reward. This formulation retains the policy's conditioning information and treats the tool response as part of the state transition.

Proof of Lemma 1. Fix h and a decision u with $\pi(u \mid h) > 0$. For every outcome r with positive probability under a continuation from h, Bayes' rule gives

$$\mathbb{P}_\pi(R = r \mid h)\frac{\mathbb{P}_\pi(U = u \mid h, R = r)}{\pi(u \mid h)} = \mathbb{P}_\pi(R = r \mid h, U = u). \tag{18}$$

Multiplying by r and summing over the possible returns proves

$$\mathbb{E}_{\tau\sim T(h,\pi)}[R(\tau)\rho_{\mathrm{HCA}}(h, u; R(\tau))] = \sum_r r\mathbb{P}_\pi(R = r \mid h, U = u) = Q^\pi(h, u). \tag{19}$$

Zero-probability return events contribute zero and need no conditional ratio. For a binary terminal reward, only $r = 1$ contributes. When $V^\pi(h) > 0$, the identity reduces to $Q^\pi(h, u) = V^\pi(h)\rho_{\mathrm{HCA}}(h, u; 1)$, proving the shared-parent ranking statement. □

For completed children with $p_\pi(c \mid n) > 0$ and $V^\pi(h(n)) > 0$, apply Bayes' rule to $C = (U, O)$ under the joint kernel $p_\pi(c \mid n)$. The corresponding ratio is

$$\rho_c = \frac{\mathbb{P}_\pi(C = c \mid h(n), R = 1)}{p_\pi(c \mid n)} = \frac{\mathbb{P}_\pi(R = 1 \mid h(n), C = c)}{\mathbb{P}_\pi(R = 1 \mid h(n))}. \tag{20}$$

Its numerator conditions on the realized observation. The action-conditioned value in the lemma averages over that observation. The practical score in Equation (7) is evaluated on the completed history, and therefore supplies a proxy for this completed-child quantity.

D.2 Exact Importance Identities

Proof of Proposition 1. For the direct tilt, substitution of $q_s(c) = p(c)s(c)/Z$ yields

$$\int q_s(c)\frac{Z}{s(c)}f(c)\,dc = \int p(c)f(c)\,dc. \tag{21}$$

The same calculation applies to discrete children by replacing integrals with sums. Support coverage ensures that all nonzero target contributions are represented.

For finite-pool selection, condition on the entire candidate pool. Then

$$\sum_{i=1}^{N} \alpha_i \frac{f(C_i)}{N\alpha_i} = \frac{1}{N} \sum_{i=1}^{N} f(C_i). \tag{22}$$

Averaging over the iid candidates gives $\mathbb{E}_p[f(C)]$. The resulting inverse-selection weight is

$$\frac{1}{N\alpha_i} = \frac{\frac{1}{N}\sum_j s(C_j)}{s(C_i)}. \tag{23}$$

For random continuation estimates $\widehat{f}_i$, the same proof holds if $\mathbb{E}[\widehat{f}_i \mid C_{1:N}, I = i] = f(C_i)$. A fresh policy continuation supplies this conditional estimate of the selected child's value. □

The finite-pool identity separates two sources of randomness. Candidate generation gives a sample from the base kernel, and score-based selection changes the conditional distribution over that sample. Its exact correction uses the mean score of all proposed candidates, including candidates whose continuations are not completed.

For the exact success-conditioned ratio, a child with zero success probability has zero proposal mass. The reward-value integral can still satisfy coverage when its integrand vanishes on those children. Recovering a general centered signal instead requires coverage for that signal's support. Positive proxy scores supply full conditional coverage within a finite candidate pool.

Two-child illustration. Suppose the rollout policy assigns probability $1/2$ to each of c_1, c_2, and their positive scores are $s(c_1) = 1$ and $s(c_2) = 3$. The tilted proposal samples them with probabilities $1/4$ and $3/4$. Its normalizer is $Z = 2$, so the inverse weights are 2 and $2/3$. For arbitrary child signals f_1, f_2, their weighted expectation is

$$\tfrac{1}{4}(2f_1) + \tfrac{3}{4}(\tfrac{2}{3}f_2) = \tfrac{1}{2}f_1 + \tfrac{1}{2}f_2. \tag{24}$$

The higher-scoring child receives more sampling effort, while each of its observations receives less weight in the base-policy estimate. The same calculation holds conditionally for a candidate pool containing these two children. The score controls the allocation of samples, and the correction preserves the expectation they estimate.

D.3 The Marginal Law of Finite-Pool Selection

Let $C_1, \ldots, C_N$ be iid from p and select an index in proportion to a positive score s. Exchangeability gives the marginal density of the selected child as

$$q_N(c) = Np(c)\,\mathbb{E}\left[\frac{s(c)}{s(c) + \sum_{j=2}^{N} s(C_j)}\right]. \tag{25}$$

To obtain this expression, sum the probability of selecting each of the N positions, condition on the value at that position, and integrate the remaining $N - 1$ candidates. In particular, $q_1 = p$. For scores bounded above and bounded away from zero, the law of large numbers and dominated convergence give $q_N(c)/p(c) \to s(c)/\mathbb{E}_p[s(C)]$ wherever $p(c) > 0$. This identifies the normalized score tilt as the large-pool limit of the selection rule.

The finite-pool correction in Appendix equation 23 is exact at every pool size. The normalized sibling backup in Equation (14) instead uses the completed children retained in the tree. That set contains original policy children and alternatives added at adaptively selected parents. Its scores and downstream signals are coupled to the tree-construction history. The direct-tilt model supplies the inverse-score form of the practical operator, and the empirical component comparison measures that operator on the resulting finite trees.

D.4 Self-Normalized Estimation and Tree Backups

Suppose $D_1, \ldots, D_m$ are iid draws from the direct tilt q_s, with $s > 0$ on the base support, $0 < Z < \infty$, and $\mathbb{E}_p|f(C)| < \infty$. The self-normalized estimator is

$$\widehat{\mu}_m = \frac{\sum_{j=1}^m f(D_j)/s(D_j)}{\sum_{j=1}^m 1/s(D_j)}. \tag{26}$$

By the strong law of large numbers, the numerator divided by m converges almost surely to $\mathbb{E}_p[f]/Z$, and the denominator divided by m converges to $1/Z$. Hence $\widehat{\mu}_m \to \mathbb{E}_p[f]$ almost surely. This is a consistency statement for the specified sampling model. At a finite sample size, self-normalization forms a ratio of random sums and generally introduces bias.

The practical tree backup has useful algebraic properties independently of that sampling model. Positive scores produce nonnegative weights summing to one, so the parent signal lies between the smallest and largest child signals. For any constants a and b,

$$\sum_c \widetilde{w}_{n,c}\big(af(c) + b\big) = a\sum_c \widetilde{w}_{n,c} f(c) + b. \tag{27}$$

Consequently, applying the same group-wide affine reward normalization before or after a fixed weighted backup gives the same result. A single-child node propagates its child's signal unchanged. When all leaf rewards agree, Equation (13) initializes and propagates a zero signal.

The change-of-measure results concern a fixed integrand and a specified selection law. The implemented learning rule combines proxy scores, a finite adaptive tree, group-relative leaf normalization, and a clipped GSPO objective. These components define the algorithm evaluated in Section 6. The propositions establish the sampling-estimation relationships that motivate the algorithm.

E Tree-construction implementation details

Two details clarify the construction in Algorithm 1. First, the practical hindsight score in Equation (7) uses a *completed* turn $n = (u, o)$. We execute every candidate against the environment before scoring it, then continue only the selected candidate to termination. One exploration step therefore costs N turn executions and a single full rollout. Second, line 3 clips the score at κ_{br}, which assigns exactly the value κ_{br} to every node that is not flagged. Taking the K smallest scores with uniform tie-breaking therefore expands the flagged nodes in increasing order of score and fills any remaining slots uniformly at random, which is the selection rule described in Section 4.

For the Qwen2.5-3B experiments, we use a rollout-matched variant of TreeHCA. Rather than first completing a fixed set of initial trajectories and branching afterward, we monitor the hindsight ratio during rollout generation. Whenever the hindsight ratio falls below a predefined threshold, we branch from that decision point and sample two alternative children. This keeps the total rollout budget comparable to GRPO while concentrating additional exploration on decisions identified as potentially consequential by the hindsight signal.

F Research assets

The experiments use the Qwen2.5 and Qwen3 model families and the seven question-answering benchmarks cited in Section 6. Dataset annotations supply reference answers for training-time hindsight scoring and evaluation. Retrieved passages enter the agent's history as tool observations and are masked from the policy-token loss. Training uses AT^{2}PO and VeRL (appendix B).

G Sample generations during training

The sibling rollouts below come from one tree during Qwen3-8B training on HotpotQA. Their shared prefix retrieves passages mentioning several NBA Finals series (Appendix table 10). The unsuccessful continuation immediately answers San Antonio Spurs and receives reward $R = 0$ (Appendix table 11). The successful continuation searches explicitly for the Miami Heat's 2012

Algorithm 2 Hindsight-guided chain construction

Require: Prompt x, reference answer $y^\star$, frozen policy π, candidate count n, maximum turns $T_{\max}$, branching threshold κ_{br}.

Initialize the trajectory with prompt x and context $\mathcal{H}_{<0}$.
for $t = 0, \ldots, T_{\max} - 1$ **do**
  Sample a decision $u_t \sim \pi(\cdot \mid \mathcal{H}_{<t})$ and execute it.
  Compute its hindsight ratio $\widehat{\rho}_{\mathrm{HCA}}(u_t)$.
  if $\widehat{\rho}_{\mathrm{HCA}}(u_t) < \kappa_{\mathrm{br}}$ **then**
    Sample n candidate decisions $\{u_t^{(i)}\}_{i=1}^{n} \sim \pi(\cdot \mid \mathcal{H}_{<t})$ and execute each candidate.
    Compute $\widehat{\rho}_{\mathrm{HCA}}^{(i)}$ for each candidate.
    Form the hindsight-guided distribution

$$p_i = \frac{\exp\left(\log \widehat{\rho}_{\mathrm{HCA}}^{(i)}\right)}{\sum_{j=1}^{n} \exp\left(\log \widehat{\rho}_{\mathrm{HCA}}^{(j)}\right)}.$$

    Sample $I \sim \mathrm{Categorical}(p_1, \ldots, p_n)$ and set $u_t \leftarrow u_t^{(I)}$.
  Append the selected decision u_t and its observation to the trajectory.
  Update the context $\mathcal{H}_{<t+1}$.
  if u_t produces a final answer **then**
    break
return the completed trajectory with its terminal verifier reward.

opponent, retrieves Oklahoma City Thunder, and receives $R = 1$ (Appendix table 12). These trajectories expose alternative decisions under the same initial evidence: answering immediately or seeking further information. TREEHCA uses their completed outcomes to compute intermediate credit through the tree.

Shared Prefix (identical in both continuations)

Question: Juwan Howard beat what team to win his first NBA Final?
Labeled Answer: Oklahoma City Thunder

<think> Okay, I need to find out which team Juwan Howard beat to win his first NBA Finals. First, I should confirm who Juwan Howard is. He was an NBA coach, right? Wait, but the question says he beat a team to win his first NBA Final. Hmm, maybe I'm mixing up some details. Let me check. </think> <search> Juwan Howard NBA coach </search>

Page 1: "Juwan Howard" Juwan Antonio Howard (born February 7, 1973) is an American former professional basketball player who is an assistant coach for the Miami Heat of the National Basketball Association (NBA). Howard formerly played for the Heat from 2010 until 2013. A one-time All-Star and one-time All-NBA power forward, he began his NBA career as the fifth overall pick in the 1994 NBA draft, selected by the Washington Bullets.
Page 2–3: *** **Omitted for brevity** ***

<think> Wait, the search result says Juwan Howard was a player, not a coach. But the original question says he beat a team to win his first NBA Final. Maybe there's confusion here. Perhaps I should search for "Juwan Howard NBA Finals" to see if he was a coach or player in the Finals. </think> <search> Juwan Howard NBA Finals </search>

Page 1: "Juwan Howard" the finals of the NCAA Division I Men's Basketball Championship in 1992 and 1993. Howard won his first NBA championship with Miami in the 2012 NBA Finals and his second NBA championship in the 2013 NBA Finals.
Page 2: "Juwan Howard" of the Fab Five to win an NBA championship, as a role player on the 2011–12 Miami Heat. Howard was on the court as time expired in the series-clinching game because the three stars of the Heat (James, Wade and Bosh) had arranged it.
Page 3: *** **Omitted for brevity** ***

<think> The first search result says that Howard won his first NBA championship with Miami in the 2012 NBA Finals, and the second says he was a role player on the 2011–12 Miami Heat. So he was on the Heat team that won the 2012 NBA Finals. But the question is who did he beat. Let me find that information. </think> <search> 2012 NBA Finals losers </search>

Page 1: "2012 NBA Finals" had previously appeared as the SuperSonics in 1996, where they lost to the Michael Jordan-led Chicago Bulls. It also marked the Miami Heat's third appearance in the NBA Finals. The Heat previously appeared in 2006 and 2011, both times against the Dallas Mavericks. It was the first NBA Finals in 14 seasons that was not held in either the states of California or Texas; the 3 teams that won the previous 13 Western Conference titles the Dallas Mavericks, the Los Angeles Lakers and the San Antonio Spurs were eliminated by Oklahoma City in that order.
Page 2: "2012 NBA Finals" straight championship, losing to the Spurs 4–1 in a rematch of the previous season's finals. The Spurs had defeated the Thunder in the Western Conference finals prior to the NBA finals. After the Heat lost the 2014 Finals, LeBron James returned to the Cleveland Cavaliers ...
Page 3: *** **Omitted for brevity** ***

***** Shared parent: the continuations below begin here *****

Table 10: Shared prefix of the sibling rollouts on HotpotQA. Blue marks model output and green marks retrieved passages. Tool-response delimiters are omitted.

Unsuccessful Continuation Terminal reward $R = 0$

***** Shared prefix omitted; see Appendix table 10 *****

<think> Wait, the first search result says the 2012 NBA Finals were between Miami Heat and the San Antonio Spurs. So, the Heat beat the Spurs. Therefore, the answer is San Antonio Spurs. </think>
Juwan Howard won his first NBA championship with the Miami Heat in the 2012 NBA Finals. The team he beat to achieve this victory was the San Antonio Spurs.
<answer> The final answer is $\boxed{\text{San Antonio Spurs}}$ </answer>

Table 11: The unsuccessful continuation from the shared parent of Appendix table 10. It issues no further query and commits to a wrong answer in a single turn.

Successful Continuation	Terminal reward $R = 1$
***** Shared prefix omitted; see Appendix table 10 *****	
<think> Hmm, the first search result says that in the 2012 NBA Finals, the Miami Heat were in the series, but it doesn't mention the opponent. The first paragraph says the Heat had previously appeared in 2006 and 2011 against the Mavericks. The Spurs were the team that beat the Heat in the 2014 Finals. So the answer isn't in the search result. Let me think again. Since Howard was on the Heat team that won the 2012 NBA Finals, but the search result is not clear, maybe I should search for "2012 NBA Finals Miami Heat opponent" directly. </think> <search> 2012 NBA Finals Miami Heat opponent </search>	
Page 1: "2012 NBA Finals" ***** Identical to Page 1 above; omitted for brevity *****	
Page 2: "2012 NBA Finals" The 2012 NBA Finals was the championship series of the 2011–12 season of the National Basketball Association (NBA), and the conclusion of the season's playoffs. The Eastern Conference champion Miami Heat defeated the Western Conference champion Oklahoma City Thunder 4 games to 1 to win their second NBA title. Heat Small forward LeBron James was named the Finals MVP.	
Page 3: "2011–12 Miami Heat season" . . . On June 21, the Miami Heat won the NBA Championship at home against the Oklahoma City Thunder,	
<think> Wait, according to this, the Miami Heat defeated the Oklahoma City Thunder in the 2012 NBA Finals. So, the answer would be the Oklahoma City Thunder. But let me check again. From the last search result, the 2012 NBA Finals were between the Miami Heat and the Oklahoma City Thunder. So, the team Juwan Howard beat was the Oklahoma City Thunder. </think>	
Juwan Howard was a player for the Miami Heat team that won the 2012 NBA Finals. The Miami Heat defeated the Oklahoma City Thunder in that series.	
<answer> The team Juwan Howard beat to win his first NBA Finals was the $\boxed{\text{Oklahoma City Thunder}}$. </answer>	

Table 12: The successful continuation from the same shared parent. It declines to answer from the ambiguous passages and reformulates the query, which retrieves an explicit statement of the required fact.