HITRA: HIERARCHICAL TRAJECTORY RERANKING FOR LONG-CONTEXT AGENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable multi-hop question answering over thousands of documents requires agents to connect sparse evidence across long observation sequences under a limited active-context budget. Existing approaches largely follow a modular trajectory-processing paradigm, addressing memory, decision evaluation, and candidate selection. Although effective within their respective roles, they leave a coupled gap: compressed states can lose recoverable hypotheses, intermediate feedback can obscure distinct decision failures, and selection lacks a jointly trained ordering over partial trajectories. To address these issues, this paper proposes Hierarchical Trajectory Reranking for Long-Context Agent (HITRA), a framework designed to jointly address the interrelated challenges of memory compression, intermediate decision evaluation, and partial-trajectory selection. Especially, selective snapshot recall equips compact recurrent rollouts with recoverable prior hypotheses, allowing later evidence to correct state errors within a bounded active context. Hierarchical credit separately normalizes planning, action, and structural-validity feedback before combining local quality with terminal utility and exploration, distinguishing decision failures while aligning updates with task success. Listwise learning explicitly orders concurrent partial trajectories, with a separable surrogate that supports microbatch training and exactly matches the full listwise gradient at the behavior policy. Together, these methods enable evidence recovery, finer-grained attribution, and listwise ranking of concurrent partial trajectories. Extensive experiments demonstrate that across context sizes ranging from 50 to 6,400 documents, HITRA maintains (near-) optimal performance across all evaluated context scales. Controlled diagnostics further demonstrate improvements in memory recovery, root-cause localization, and trajectory selection: Compared to all baselines, the memory recovery rate increases by at least 8.7 percentage points (see Table 2 Recovery Rate), the root-cause localization rate increases by at least 9.7 percentage points (see Table 3 Root@1), and the optimal candidate selection capability improves by at least 18.0 percentage points (see Table 4 Hit@1).

1 INTRODUCTION

Reliable multi-hop question answering over thousands of documents requires agents to connect sparse evidence across long observation sequences under a limited active-context budget. Existing approaches largely follow a modular trajectory-processing paradigm, addressing memory compression (Yu et al., 2026; Xu et al., 2025; Jiménez Gutiérrez et al., 2025), decision evaluation (Weston et al., 2025; Zhou et al., 2025; Feng et al., 2025), and candidate selection (Putta et al., 2024; Chae et al., 2025) as separate problems. Although effective within their respective roles, this decoupling leaves a coupled gap: compressed states can lose recoverable hypotheses, intermediate feedback can obscure distinct decision failures, and selection lacks a jointly trained ordering over partial trajectories during long-context reasoning.

Recent work tackles each stage largely in isolation. For memory, MemAgent (Yu et al., 2026) trains a memory agent with multi-conversation RL to read text segment by segment and overwrite a bounded memory, extrapolating to million-token inputs but irreversibly discarding the raw evidence behind pruned content. A-MEM (Xu et al., 2025) organizes experience into Zettelkasten-style notes with dynamic linking and memory evolution, preserving associative structure while retaining only distilled notes rather than recoverable observations. Memory-R1 (Yan et al., 2025) fine-tunes a memory manager and an answer agent with outcome-driven RL to learn add/update/delete operations, optimizing what to keep for answer accuracy rather than the recoverability of discarded states.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

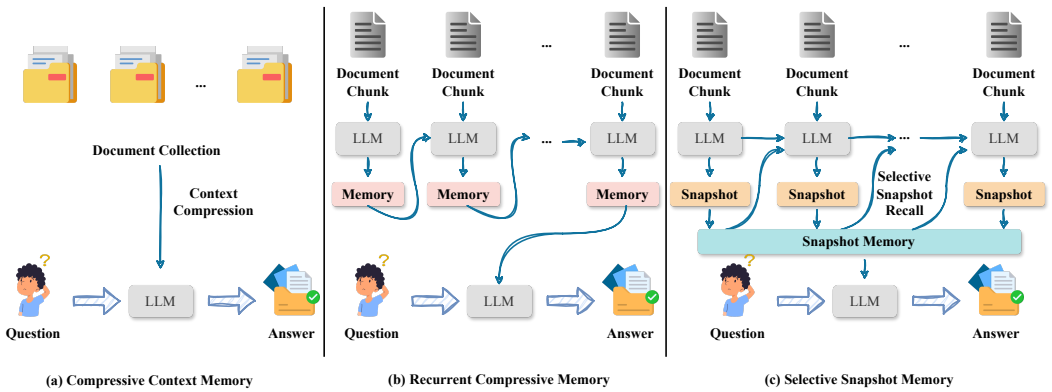

Figure 1: Comparison between HITRA and two representative methods. (a) Compressive context memory condenses the document collection before answering and may discard useful information. (b) Recurrent compressive memory propagates a compact state, making omitted earlier information unrecoverable. (c) HITRA adds private trajectory snapshots and selective recall to recurrent processing, allowing later evidence to revise earlier interpretations under a bounded context.

For decision evaluation, StepWiser (Weston et al., 2025) reframes stepwise reward modeling as meta-reasoning, training a generative judge by RL to critique intermediate reasoning chunks before issuing a verdict. SWEET-RL (Zhou et al., 2025) trains an asymmetric critic with access to training-time information to emit step-level rewards for multi-turn agents, yet each scalar still aggregates heterogeneous error sources within a step. GiGPO (Feng et al., 2025) estimates two-level group-relative advantages over episode groups and anchor-state step groups for fine-grained per-step credit without a critic, attributing value to steps but not to the type of decision that failed.

For candidate selection, Agent Q (Putta et al., 2024) combines guided MCTS with self-critique and iterative DPO to expand and prune web-agent branches, relying on search statistics and pairwise preferences rather than a jointly trained ordering. WMA (Chae et al., 2025) equips a web agent with a world model that simulates candidate action outcomes to pick the next step, performing selection purely at inference time without a trained trajectory scorer. RankRAG (Yu et al., 2024) instruction-tunes a single LLM to both rank retrieved contexts and generate answers, coupling ranking with generation only over complete documents rather than concurrent partial trajectories.

To address the above limitations, this paper proposes HITRA, a framework that jointly addresses memory compression, intermediate decision evaluation, and partial-trajectory selection for long-context agents. By coupling selective snapshot recall, hierarchical credit assignment, and listwise trajectory learning, HITRA supports evidence recovery, fine-grained attribution, and learned trajectory ranking under a bounded active context (Figure 1). The main contributions are as follows:

The HITRA framework. This paper proposes Hierarchical Trajectory Reranking for Long-Context Agent (HITRA) for multi-hop question answering over thousands of documents under a limited active-context budget. The framework jointly addresses the interrelated problems of evidence loss during memory compression, ambiguous intermediate decision feedback, and the lack of a jointly trained ordering over partial trajectories by coupling memory recovery, hierarchical credit assignment, and listwise trajectory learning.

Selective snapshot recall. Selective snapshot recall equips compact recurrent rollouts with recoverable prior hypotheses, allowing later evidence to correct state errors under a bounded active-context budget. This mechanism enables evidence recovery and hypothesis revision without retaining the full observation history in the active context.

Hierarchical credit assignment. Hierarchical credit separately normalizes planning, action, and structural-validity feedback before combining local quality with terminal utility and exploration. This design distinguishes different sources of decision failure and provides finer-grained credit attribution while aligning policy updates with task success.

Listwise trajectory learning. A listwise learning objective explicitly orders concurrent partial trajectories, enabling candidate selection through a jointly trained ranking. A separable surrogate supports microbatch training while exactly matching the full listwise gradient at the behavior policy, preserving the listwise training signal when candidates are processed in separate microbatches.

Experimental results. Extensive experiments demonstrate that HITRA maintains optimal or near-optimal performance across context sizes from 50 to 6,400 documents, reaching 81.3% accuracy at the largest scale. Controlled diagnostics show gains over all evaluated baselines of at least 8.7 percentage points in memory recovery, 9.7 percentage points in root-cause localization, and 18.0 percentage points in optimal-candidate selection (Hit@1).

2 Related Work

Long-context sequence and memory models. Transformer-XL, Compressive Transformer, and Recurrent Memory Transformer carry hidden states, compressed activations, or learned memory tokens across segments (Dai et al., 2019; Rae et al., 2020; Bulatov et al., 2022). MemGPT moves records between archival storage and active context (Packer et al., 2023), but accessible history alone does not ensure reliable use of dispersed evidence (Liu et al., 2024). HITRA, for the first time, adds private snapshot recall to bounded recurrent processing so that later evidence can trigger revision of an earlier hypothesis during the reasoning process.

Reasoning and intermediate credit. Chain-of-thought, self-consistency, and ReAct expose reasoning, aggregate samples, and ground actions (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023b). Reflexion and Self-Refine use iterative feedback, while Tree of Thoughts and LATS preserve alternatives through search (Shinn et al., 2023; Madaan et al., 2023; Yao et al., 2023a; Zhou et al., 2024). RLHF, PPO, and GRPO support preference-based or relative policy updates (Ouyang et al., 2022; Schulman et al., 2017; Shao et al., 2024), process supervision and Agent Lightning provide intermediate credit (Lightman et al., 2024; Luo et al., 2025). HITRA separately normalizes planning, action, and validity feedback within same-task, same-step cohorts, then combines it with terminal utility and exploration.

Listwise reranking and preference optimization. ListNet fits a distribution over candidates (Cao et al., 2007); RRHF and DPO learn from ranked or paired responses (Yuan et al., 2023; Rafailov et al., 2023). HITRA instead ranks synchronized partial trajectories using reward and exploration targets. Detached per-turn weights match the full listwise gradient at the behavior policy, allowing microbatch training of the same recurrent policy that generates and scores trajectories.

3 Problem Formulation

Task and recurrent rollout. A task q contains ordered observation chunks $\mathcal{C}_q = (c_{q,1}, \ldots, c_{q,N_q})$, with $N_q \geq 1$. The fixed behavior policy $\pi_{\theta_{\text{old}}}$ independently samples $K \geq 1$ replicas indexed by k. For a fixed task, let $N = N_q$, $c_t = c_{q,t}$ and omit q from recurrent subscripts.

Replica k carries compact state z_t^k, initialized by $z_0^k = z_{\text{init}}(q)$, and a private snapshot store $\mathcal{M}_t^k$ available before step t. The optional recalled snapshot is m_t^k, initially $\mathcal{M}_1^k = m_1^k = \varnothing$. Its input and turn are $s_t^k = (q, c_t, z_{t-1}^k, m_t^k)$ and $\tau_t^k \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_t^k)$, where $m_t^k \in \mathcal{M}_t^k \cup \{\varnothing\}$ and $k \in \{1, \ldots, K\}$. A turn records a reasoning summary, plan, action or retrieval decision, evidence, compact-state update, and optional recall query. The $t = N + 1$ is reserved for final answering: the input s_{N+1}^k produces replica answer $\hat{y}^k$ as

$$s_{N+1}^k = (q, z_N^k, m_{N+1}^k), \quad \tau_{N+1}^k = \hat{y}^k \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_{N+1}^k). \tag{1}$$

Synchronized cohorts. The synchronized cohort is $\mathcal{G}_{q,t} = \big((s_t^k, \tau_t^k)\big)_{k=1}^K$ for $t \in \{1, \ldots, N_q + 1\}$.

The index preserves duplicate samples. Cohort members share the task, chunk, and recurrent depth but may have different histories after $t = 1$, comparisons measure relative trajectory progress, not counterfactual actions from one state.

Inference-time selection. At inference, the trained actor π_θ replaces the behavior policy. With length-normalized intermediate-turn score $\ell_{q,t,k}(\theta)$, the answer with the highest mean score is returned: $S_q^k(\theta) = \frac{1}{N_q} \sum_{t=1}^{N_q} \ell_{q,t,k}(\theta)$, and $k^\star \in \arg\max_{1 \leq k \leq K} S_q^k(\theta)$, and $\hat{y} = \hat{y}^{k^\star}$. Steps have equal weight and final-answer likelihood is excluded. Selection needs neither rewards nor a separate reranker; comparisons match rollout and token budgets.

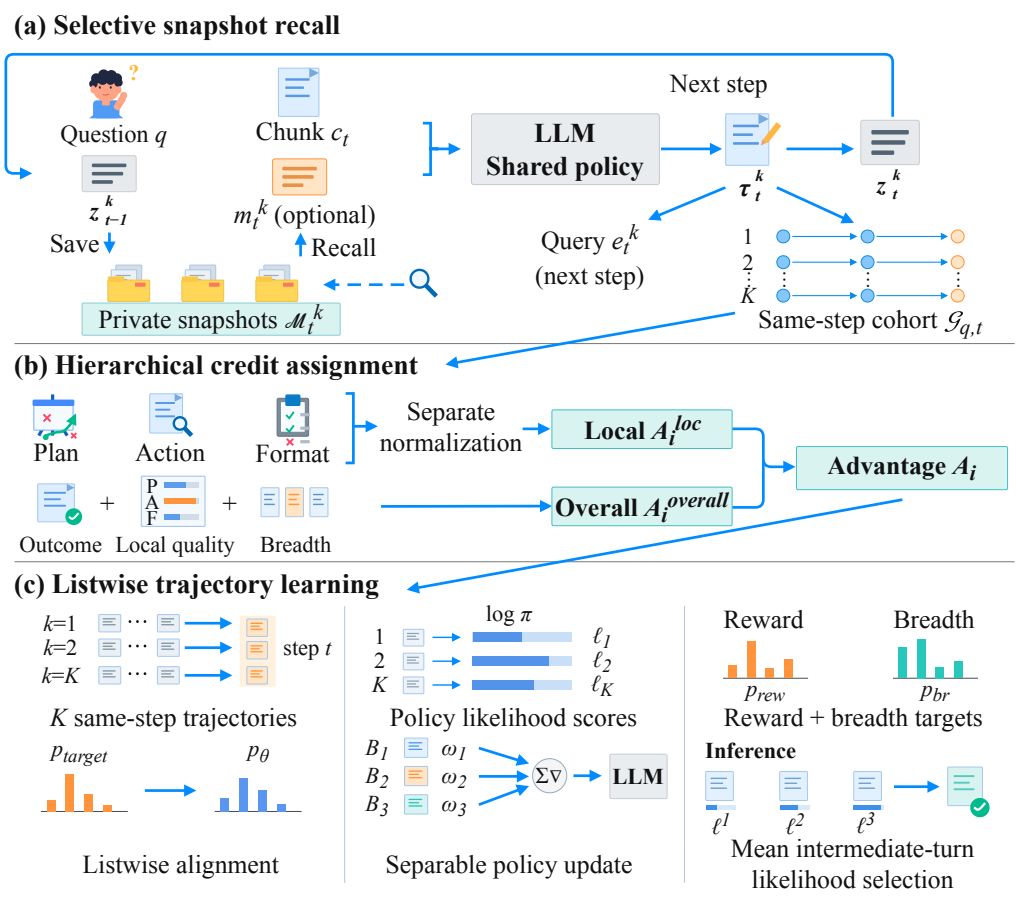

Figure 2: Overview of HITRA. Each rollout processes document chunks with a compact state and optional selective snapshot recall. Synchronized partial trajectories provide hierarchical credit and listwise ranking signals, the trained shared policy selects an answer using mean intermediate-turn likelihood at inference time.

4 Proposed Method

HITRA couples selective snapshot recall, hierarchical credit, and listwise learning in a shared recurrent policy (Figure 2). Each of K replicas processes a chunk with a compact state and optional private snapshot. Its generated turn updates the state and may retrieve an earlier hypothesis for the next turn. Same-task, same-step replicas form a cohort for normalizing local feedback and constructing reward- and exploration-induced ranking targets. A joint actor, Kullback-Leibler (KL), and ranking objective trains the policy; detached ranking weights permit microbatch accumulation while matching the full-cohort gradient at the behavior policy. At inference, mean intermediate-turn likelihood selects the answer.

4.1 Selective snapshot recall

Selective snapshot recall preserves prior hypotheses outside the active context so later evidence can correct state errors. At step t, replica k carries state z^k_{t-1} and private store $\mathcal{M}^k_t$. A fixed parser extracts updated state z^k_t and optional query e^k_t; a uniquely indexed snapshot ν^k_t stores the pre-update state and provenance. The fixed retriever returns next-turn payload m^k_{t+1}: $\mathrm{Parse}(\tau^k_t) = (z^k_t, e^k_t)$, and $\mathcal{M}^k_{t+1} = \mathcal{M}^k_t \cup \{\nu^k_t\}$, and $m^k_{t+1} = \mathrm{Retrieve}(e^k_t, \mathcal{M}^k_t)$. Retrieval returns one snapshot from the pre-step store $\mathcal{M}^k_t$, or $\varnothing$ if the query or store is empty. The new ν^k_t becomes eligible only at the next step, preventing self-matches. The payload augments the next prompt without overwriting the state or rewinding the document stream, the policy can compare it with new evidence and revise its interpretation. Final-step recall can inform Eq. (1).

Predecessor links record chronological states and recall links identify revisited hypotheses; private stores prevent cross-trajectory information transfer. For enforced token budgets $B_q, B_c, B_z, B_m, B_{\text{inst}}$ on the question, chunk, state, recalled payload, and instructions plus serialization, $|s_t^k|_{\text{tok}} \le B_q + B_c + B_z + B_m + B_{\text{inst}}$. Here $|\cdot|_{\text{tok}}$ counts tokens. Active context is independent of processed length, but external storage is not: without eviction, each replica retains $\Theta(N_q)$ snapshots with $O(N_q B_z)$ state-payload tokens.

Why selective snapshot recall is useful. External snapshots preserve hypotheses omitted from the compact state, allowing later evidence to trigger reconsideration. Recall supplements the current state without rewinding the document stream, supporting correction of compression-induced errors under a bounded active context, while external storage grows with the processed sequence.

4.2 Hierarchical credit assignment

Local credit distinguishes planning, action-grounding, and structural errors, trajectory credit links them to terminal success and exploration. Both compare alternatives from the same task and recurrent step within each synchronized cohort.

Cohort-relative normalization. For turn $i = (q, t, k)$, let $\mathcal{G}(i) = \mathcal{G}_{q,t}$. Each feedback channel x has cohort mean $\mu_{x,\mathcal{G}} = K^{-1} \sum_k x_{q,t,k}$ and population variance $\sigma^2_{x,\mathcal{G}} = K^{-1} \sum_k (x_{q,t,k} - \mu_{x,\mathcal{G}})^2$. With standard deviation $\sigma_{x,\mathcal{G}}$, stabilizer $\varepsilon_{\text{num}} > 0$, and stop-gradient operator sg, define the detached normalization $\mathcal{A}_\delta(x_i; \mathcal{G}) = \text{sg}\left[\frac{x_i - \mu_{x,\mathcal{G}}}{(1-\delta) + \delta(\sigma_{x,\mathcal{G}} + \varepsilon_{\text{num}})}\right]$. Here $\delta = 0$ mean-centers each channel and $\delta = 1$ standardizes it. Credits sum to zero and vanish for constant signals. Cohorts provide a same-task, same-step reference without a value network; differing histories make this relative progress rather than counterfactual action value.

Semantic-level local credit. Each intermediate turn receives plan, action, and format scores $r_i^{\text{plan}}, r_i^{\text{act}}, r_i^{\text{fmt}}$. The plan score assesses subgoal relevance, state consistency, and evidence progress; the action score assesses validity and grounding in the observation or recalled snapshot. Format checks deterministically verify required fields. Task verifiers or a frozen rubric-based scorer supply semantic scores, never the trainable actor; all rewards are detached.

Let $\mathcal{D}_{\text{loc}} = \{\text{plan}, \text{act}, \text{fmt}\}$ be the set of local feedback dimensions. For each $\eta \in \mathcal{D}_{\text{loc}}$, choose a non-negative weight w_η, and let $W_{\text{loc}} = \sum_{\eta \in \mathcal{D}_{\text{loc}}} w_\eta > 0$ be their total. Set $A_i^\eta = \mathcal{A}_\delta(r_i^\eta; \mathcal{G}(i))$.

The local reward and local advantage are distinct quantities: $r_i^{\text{loc}} = \sum_{\eta \in \mathcal{D}_{\text{loc}}} \frac{w_\eta}{W_{\text{loc}}} r_i^\eta$, and $A_i^{\text{loc}} = \sum_{\eta \in \mathcal{D}_{\text{loc}}} \frac{w_\eta}{W_{\text{loc}}} A_i^\eta$. Separate normalization distinguishes failure types and, for $\delta = 1$, limits high-variance channel dominance. The raw mixture r_i^{loc} enters trajectory utility; A_i^{loc} supplies direct local credit. Both are zero on final turns.

Trajectory-level overall credit. Let $r_{q,k}^{\text{out}}$ be the terminal task score of replica k for task q, and $r_{q,k}^{\text{ansfmt}}$ its answer-format score. Define $\iota_i = \mathbb{I}\{t \le N_q\}$, where $\mathbb{I}\{\cdot\}$ is the indicator function, to distinguish intermediate from final turns. Terminal scores are broadcast only along the same (q, k) rollout to its constituent turns.

For an intermediate turn, let h_i^{depth}, h_i^{novel}, and h_i^{resolve} denote fixed scores for underused hypotheses, non-redundant grounded evidence, and reduction of explicit uncertainty, respectively. Each component is scaled to $[-1, 1]$. Given non-negative coefficients β_{dep}, β_{nov}, and β_{res}, define the detached breadth signal $h_i^{\text{breadth}} = \text{sg}\left[\beta_{\text{dep}} h_i^{\text{depth}} + \beta_{\text{nov}} h_i^{\text{novel}} + \beta_{\text{res}} h_i^{\text{resolve}}\right]$. The bound is $|h_i^{\text{breadth}}| \le \beta_{\text{dep}} + \beta_{\text{nov}} + \beta_{\text{res}}$. Reproduction requires specifying the node-use rule, evidence similarity, uncertainty evaluator, and scaling constants. Set $h_i^{\text{breadth}} = 0$ on final turns. This exploration prior need not preserve the unshaped optimum.

Let w_{out}, w_{loc}, w_{br}, and w_{af} be non-negative weights for terminal outcome, local quality, breadth, and answer-format quality, respectively. The overall utility R_i is $R_i = w_{\text{out}} r_{q,k}^{\text{out}} + \iota_i \big(w_{\text{loc}} r_i^{\text{loc}} + w_{\text{br}} h_i^{\text{breadth}}\big) + (1 - \iota_i) w_{\text{af}} r_{q,k}^{\text{ansfmt}}$. Final turns retain only outcome and answer-format utility.

Broadcasting terminal scores aligns updates with task success without asserting causal responsibility for every earlier action.

Overall utility is normalized within the corresponding intermediate or final cohort and then combined with semantic-level credit. Let $\alpha \in [0,1]$ control this mixture. Define $A_i^{\text{overall}} = \mathcal{A}_\delta(R_i; \mathcal{G}(i))$, and $A_i = \text{sg}\big[A_i^{\text{overall}} + \iota_i(1-\alpha)\big(A_i^{\text{loc}} - A_i^{\text{overall}}\big)\big]$. Intermediate turns mix local and overall credit; final turns receive their full cohort advantage. Increasing α emphasizes overall utility, which includes local and breadth terms rather than terminal reward alone. The masks separate intermediate feedback from final-answer formatting.

This hierarchy distinguishes planning, acting, and formatting failures that terminal rewards or a single process score can obscure. Local credit identifies the failure type, while trajectory utility connects updates to final-answer quality and exploration.

Why hierarchical credit assignment is useful. Separate feedback channels distinguish planning, grounding, and formatting failures, while cohort-relative normalization provides a comparable reference. Combining local credit with trajectory utility links intermediate decisions to task success and exploration, without assuming that a successful outcome makes every preceding decision correct.

4.3 Listwise trajectory learning

To turn hierarchical evaluation into selection, the policy's likelihood distribution over each cohort is matched to reward and breadth targets. A separable surrogate supports joint training with the token-level actor and KL losses.

Policy-induced trajectory distribution. For every retained turn $i = (q,t,k)$, let $n_i \geq 1$ be the number of generated response tokens, excluding prompt and padding. Let s_i and τ_i denote the actor input and generated turn indexed by i; $\tau_{i,j}$ is its jth token, $\tau_{i,<j}$ is the preceding token prefix, and $\xi_{i,j} = (s_i, \tau_{i,<j})$ is the recorded context for that token. The shared policy assigns the turn the length-normalized score $\ell_i(\theta) = \frac{1}{n_i}\sum_{j=1}^{n_i} \log \pi_\theta(\tau_{i,j} \mid \xi_{i,j})$. Length normalization avoids favoring longer responses. Recorded states and prefixes are fixed, so gradients do not pass through sampling or retrieval. For intermediate cohort $\mathcal{G}_{q,t}$, abbreviate $\ell_k(\theta) = \ell_{q,t,k}(\theta)$.

Credit-induced listwise targets. Let $\vartheta_\pi > 0$ be the policy temperature and let $\vartheta_{\text{rew}} > 0$ and $\vartheta_{\text{br}} > 0$ be the reward and breadth temperatures. The policy scores induce the cohort distribution $p_\theta(k \mid \mathcal{G}) = \frac{\exp(\ell_k(\theta)/\vartheta_\pi)}{\sum_{k'=1}^{K}\exp(\ell_{k'}(\theta)/\vartheta_\pi)}$. Let $\gamma \in \{\text{rew}, \text{br}\}$ identify the ranking signal and define its candidate value by $\zeta_k^{\text{rew}} = R_{q,t,k}$ and $\zeta_k^{\text{br}} = h_{q,t,k}^{\text{breadth}}$. Overall reward and breadth then induce two detached target distributions: $p_\gamma(k \mid \mathcal{G}) = \text{sg}\Big[\frac{\exp(\zeta_k^\gamma/\vartheta_\gamma)}{\sum_{k'=1}^{K}\exp(\zeta_{k'}^\gamma/\vartheta_\gamma)}\Big]$, $\gamma \in \{\text{rew}, \text{br}\}$. All candidate sums stay within one task and step. The policy is matched to each target with the top-one ListNet objective defined over the same intermediate cohort

$$\mathcal{L}_\gamma^{\mathcal{G}}(\theta) = -\sum_{k=1}^{K} p_\gamma(k \mid \mathcal{G}) \log p_\theta(k \mid \mathcal{G}), \qquad \gamma \in \{\text{rew}, \text{br}\}. \tag{2}$$

These probabilities encode a cohort ordering without enumerating $K!$ permutations. Reward already includes breadth when $w_{\text{br}} > 0$; the separate breadth target adds independently controlled exploration pressure within the same cohort.

Separable gradient-matching surrogate. The softmax denominator in Eq. (2) couples all K candidates, which conflicts with length-based sharding and microbatch accumulation. The full-cohort statistics are therefore computed once at the behavior policy, $\ell_k^{\text{old}} = \text{sg}[\ell_k(\theta_{\text{old}})]$ and $p_{\text{old}}(k \mid \mathcal{G}) = \text{sg}[p_{\theta_{\text{old}}}(k \mid \mathcal{G})]$. Differentiating with respect to ℓ_k gives $\frac{\partial \mathcal{L}_\gamma^{\mathcal{G}}}{\partial \ell_k} = \frac{p_\theta(k|\mathcal{G}) - p_\gamma(k|\mathcal{G})}{\vartheta_\pi}$. This gradient yields detached cohort weights and a centered linear surrogate:

$$\omega_k^\gamma = \text{sg}\big[K\big(p_\gamma(k \mid \mathcal{G}) - p_{\text{old}}(k \mid \mathcal{G})\big)\big], \qquad \sum_{k=1}^{K} \omega_k^\gamma = 0, \tag{3}$$

Zero-sum weights favor target-preferred candidates without a uniform shift. Detachment makes their linear surrogate separable: $\widetilde{\mathcal{L}}_\gamma^{\mathcal{G}}(\theta) = \frac{1}{K\vartheta_\pi}\sum_{k=1}^{K}\omega_k^\gamma\big[\ell_k^{\text{old}} - \ell_k(\theta)\big], \gamma \in \{\text{rew}, \text{br}\}$. The factor K compensates for candidate averaging. Consequently, the surrogate reproduces the scale and direction of the full listwise gradient at the behavior policy:

$$\nabla_\theta \widetilde{\mathcal{L}}_\gamma^{\mathcal{G}}(\theta)\Big|_{\theta=\theta_{\text{old}}} = \nabla_\theta \mathcal{L}_\gamma^{\mathcal{G}}(\theta)\big|_{\theta=\theta_{\text{old}}}. \tag{4}$$

This is a first-order guarantee: the surrogate is zero at θ_{old} and matches the listwise gradient there, but not necessarily after an update. Appendix E derives the exact remainder. Targets and weights are recomputed per rollout batch and fixed through its update epochs; final turns are excluded from ranking within the intermediate cohorts.

Joint optimization objective. Let $\mathcal{Q}$ be the task batch, $\mathcal{I}$ its retained turns, $N_{\text{tok}} = \sum_{i\in\mathcal{I}} n_i$ the response-token count, and $\mathfrak{G} = \{\mathcal{G}_{q,t} : q \in \mathcal{Q}, 1 \le t \le N_q\}$ the intermediate cohorts, with $N_{\text{coh}} = \sum_q N_q$. The token likelihood ratio on recorded context $\xi_{i,j}$ is $\rho_{i,j}(\theta) = \frac{\pi_\theta(\tau_{i,j}|\xi_{i,j})}{\pi_{\theta_{\text{old}}}(\tau_{i,j}|\xi_{i,j})}$. The denominator is detached. With $\varepsilon_{\text{clip}} \in (0,1)$, write $\bar{\rho}_{i,j} = \text{clip}(\rho_{i,j}, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}})$.

The clipped ratio yields the following token-averaged actor loss for policy optimization $\mathcal{L}_{\text{clip}}(\theta) = -\frac{1}{N_{\text{tok}}}\sum_{i\in\mathcal{I}}\sum_{j=1}^{n_i}\min\{\rho_{i,j}(\theta)A_i,\ \bar{\rho}_{i,j}(\theta)A_i\}$. All tokens in turn i share A_i; reward factorization does not assign separate advantages to plan and action spans. For frozen reference policy π_{ref} with support on vocabulary $\mathcal{V}$, define $\text{KL}_{i,j}(\theta) = \sum_{v\in\mathcal{V}}\pi_\theta(v \mid \xi_{i,j})\log\frac{\pi_\theta(v|\xi_{i,j})}{\pi_{\text{ref}}(v|\xi_{i,j})}$. Average this conditional policy divergence over recorded contexts:

$$\mathcal{L}_{\text{KL}}(\theta) = \frac{1}{N_{\text{tok}}}\sum_{i\in\mathcal{I}}\sum_{j=1}^{n_i}\text{KL}_{i,j}(\theta). \tag{5}$$

Contexts are behavior-generated, but the vocabulary expectation uses the current actor; a sampled-token estimator is not identical to this KL.

Finally, average each ranking loss over intermediate cohorts so that its scale is independent of response length: $\overline{\mathcal{L}}_\gamma(\theta) = \frac{1}{N_{\text{coh}}}\sum_{\mathcal{G}\in\mathfrak{G}}\widetilde{\mathcal{L}}_\gamma^{\mathcal{G}}(\theta), \gamma \in \{\text{rew}, \text{br}\}$. With non-negative coefficients $\lambda_{\text{KL}}, \lambda_{\text{rew}}, \lambda_{\text{br}}$, the joint objective is

$$\mathcal{L}_{\text{HITRA}}(\theta) = \mathcal{L}_{\text{clip}}(\theta) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}(\theta) + \lambda_{\text{rew}}\overline{\mathcal{L}}_{\text{rew}}(\theta) + \lambda_{\text{br}}\overline{\mathcal{L}}_{\text{br}}(\theta). \tag{6}$$

Actor and KL terms cover all turns; ranking covers intermediate cohorts with equal weight, so longer tasks contribute more cohorts. The shared policy learns both generation and selection, while parsing and retrieval remain fixed.

Why listwise trajectory learning is useful. Matching policy scores to reward and breadth targets connects trajectory evaluation to candidate selection. Length normalization avoids favoring longer turns, and the separable surrogate supports microbatch training while matching the listwise gradient at the behavior policy. Inference requires neither rewards nor a separate reranker.

5 Experiments

HITRA is evaluated on the HotpotQA distractor setting (Yang et al., 2018), using the official training split for optimization and disjoint development examples for model selection and reporting. Primary training and diagnostic experiments use Qwen2.5-3B-Instruct, while the scaling study additionally evaluates Qwen2.5-7B-Instruct (Qwen et al., 2024). The implementation uses HybridFlow/`verl` with fully sharded data parallel training (Sheng et al., 2025) and SGLang for recurrent generation (Zheng et al., 2024). Each batch contains 128 questions with $K = 8$ replicas, contexts are processed in at most eight 5,000-token chunks, and each trajectory or final answer is capped at 2,048 tokens. Models train for five epochs with $\alpha = 0.8$, $\varepsilon_{\text{clip}} = 0.2$, $\lambda_{\text{KL}} = 10^{-3}$, and an actor learning rate of 10^{-6}. Appendix C provides the complete implementation, evaluation, baseline, and ablation settings.

Long-context scaling. Table 1 evaluates 50–6,400 documents. HITRA leads at six of eight 3B settings and all 7B settings, reaching 81.3% at 6,400 documents. At 3B/6,400, it scores 65.7% versus the best baseline's 66.1%. Qwen2.5-1M drops from 75.8% to 0.0%. These results highlight HITRA's ability to sustain competitive accuracy and reliable evidence use as document collections grow.

Table 1: HotpotQA accuracy (%) as the number of context documents grows. Results are grouped by model scale. A dash denotes an unavailable evaluation. Best and second-best results are bolded and underlined, respectively.

	Number of context documents							
Method	50	100	200	400	800	1,600	3,200	6,400
(a) 3B models								
Qwen2.5	59.4	57.0	—	—	—	—	—	—
MemAgent	70.3	69.4	60.9	68.8	60.9	60.2	59.4	58.8
ReMemR1	<u>70.9</u>	<u>71.7</u>	<u>63.8</u>	**74.0**	<u>65.4</u>	<u>65.0</u>	<u>65.4</u>	**66.1**
HITRA (ours)	**75.1**	**74.8**	**67.6**	<u>71.2</u>	**67.0**	**66.3**	**66.1**	<u>65.7</u>
(b) 7B models								
Qwen2.5	70.3	75.0	—	—	—	—	—	—
R1-Distill	40.6	25.8	10.2	0.8	1.6	2.3	1.5	3.1
Qwen2.5-1M	75.8	71.9	68.0	67.2	69.5	54.7	22.7	0.0
MemAgent	81.8	78.9	78.9	77.0	79.7	72.1	74.0	75.8
ReMemR1	<u>82.3</u>	<u>82.8</u>	<u>81.1</u>	<u>78.9</u>	<u>82.0</u>	<u>79.7</u>	<u>80.0</u>	<u>80.8</u>
HITRA (ours)	**85.7**	**86.1**	**84.9**	**81.0**	**82.2**	**80.5**	**80.6**	**81.3**

Recoverability of snapshot memory. On controlled state corruption (Table 2), HITRA leads snapshot Hit@1 (71.6%) and recovery rate (63.4%). Against similarity retrieval, it gains 6.3 percentage points in Answer F1 and 6.1 percentage points in Joint F1 with 130 fewer recalled tokens. Random recall underperforms no recall. These results highlight selective snapshot recall's ability to recover useful prior evidence and correct corrupted states while limiting recall cost.

Table 2: Memory recovery on 1,000 controlled HotpotQA state-corruption examples with Qwen2.5-3B-Instruct. The two panels report recovery behavior and downstream quality/efficiency for the same methods. Best and second-best results are bolded and underlined, respectively.

(a) Recovery behavior (first four metrics in %; recovery steps in counts)

Method	Snapshot Hit@1↑	Harmful Recall↓	State Correction↑	Recovery Rate↑	Recovery Steps↓
No Recall	—	—	15.1	20.6	8.5
Recent Snapshot	34.8	29.3	31.4	35.3	7.0
Random Snapshot	14.3	41.5	17.7	22.9	8.2
Similarity Retrieval	<u>58.1</u>	<u>16.5</u>	<u>49.8</u>	<u>54.7</u>	<u>5.8</u>
HITRA(ours)	**71.6**	**7.9**	**58.4**	**63.4**	**4.2**

(b) Task quality and recall cost

Method	Answer F1 (%)↑	Joint F1 (%)↑	Recall Tokens↓
No Recall	42.7	29.8	**0**
Recent Snapshot	45.8	32.1	456
Random Snapshot	41.6	27.6	459
Similarity Retrieval	<u>52.2</u>	<u>39.4</u>	448
HITRA(ours)	**58.5**	**45.5**	<u>318</u>

Factorized credit assignment. Table 3 tests localization of a verified corrupted step and component. Compared with unfactorized credit, HITRA gains 9.7 percentage points in Root@1, raises counterfactual correlation from 0.558 to 0.681, and reduces leakage by 11.8 percentage points while attaining the highest Joint F1 (60.6%). These results highlight factorized credit assignment's ability to distinguish decision failures and reduce attribution leakage, supporting more effective credit assignment and downstream answering.

Listwise trajectory selection. On shared $K = 8$ candidates (Table 4), HITRA reaches 47.7% Hit@1 and 0.883 NDCG@8. Against recurrent GRPO scoring, it gains 18.0 percentage points in Hit@1 and 6.5 percentage points in Joint F1, reducing regret from 17.6 to 11.1. Both ranking targets help; removing reward ranking hurts more, while breadth adds complementary coverage. Together, the two targets enable listwise trajectory learning to select higher-quality candidates and improve downstream answer quality.

Table 3: Credit-assignment accuracy on the controlled counterfactual intervention set. Results are mean±standard deviation over three independent training seeds. Best and second-best results are bolded and underlined, respectively.

(a) Root-cause localization

Method	Root@1 (%)↑	MRR (%)↑	AUROC (%)↑
Recurrent GRPO (outcome only)	18.7±1.6	34.9±1.3	62.1±1.1
Scalar process reward	35.1±1.7	51.2±1.4	73.6±1.0
HITRA w/o local credit	29.4±1.5	45.8±1.3	69.2±1.1
HITRA w/ unfactorized local credit	<u>46.8±1.4</u>	<u>61.7±1.2</u>	<u>81.5±0.8</u>
HITRA (ours)	**56.5±1.2**	**70.1±1.0**	**87.0±0.7**

(b) Component attribution, causal fidelity, and task quality (percentage metrics in %)

Method	Sem.-F1↑	Fmt.-F1↑	ρ_{CF} ↑	Leakage↓	Joint F1↑
Recurrent GRPO (outcome only)	43.2±1.8	60.5±1.4	0.238±0.031	67.8±2.0	52.4±0.8
Scalar process reward	56.7±1.6	71.4±1.3	0.421±0.027	49.6±1.8	54.6±0.7
HITRA w/o local credit	50.1±1.7	66.2±1.4	0.354±0.030	56.9±1.9	56.3±0.6
HITRA w/ unfactorized local credit	<u>65.4±1.4</u>	<u>79.3±1.1</u>	<u>0.558±0.022</u>	<u>36.7±1.5</u>	<u>58.2±0.6</u>
HITRA (ours)	**74.6±1.1**	**87.2±0.8**	**0.681±0.018**	**24.9±1.2**	**60.6±0.5**

Table 4: Trajectory selection on a shared batch of $K = 8$ candidate trajectories. The two panels separate ranking quality from downstream answer and evidence quality. Best and second-best results are bolded and underlined, respectively.

(a) Ranking and headroom recovery

Selector	Hit@1 (%)↑	NDCG@8↑	p_{gold} ↑	Regret↓	HRC (%)↑
Random candidate	12.5	0.705	0.00	23.0	0.0
Self-consistency	21.0	0.755	0.15	20.9	9.1
Final-answer likelihood	18.6	0.742	0.11	21.7	5.7
Recurrent GRPO score (no listwise loss)	29.7	0.801	0.29	17.6	23.5
HITRA w/o reward ranking ($\lambda_{\mathrm{rew}} = 0$)	34.1	0.820	0.36	16.0	30.4
HITRA w/o breadth ranking ($\lambda_{\mathrm{br}} = 0$)	<u>39.3</u>	<u>0.846</u>	<u>0.43</u>	<u>14.2</u>	<u>38.3</u>
HITRA (ours)	**47.7**	**0.883**	**0.55**	**11.1**	**51.7**

(b) Downstream task quality (%)

Selector	Answer EM ↑	Answer F1 ↑	Support F1 ↑	Joint F1 ↑
Random candidate	40.2	53.8	61.2	36.0
Self-consistency	42.6	56.4	61.8	38.1
Final-answer likelihood	41.5	55.2	61.2	37.3
Recurrent GRPO score (no listwise loss)	45.1	59.1	64.5	41.4
HITRA w/o reward ranking ($\lambda_{\mathrm{rew}} = 0$)	46.0	60.0	66.0	43.0
HITRA w/o breadth ranking ($\lambda_{\mathrm{br}} = 0$)	<u>47.6</u>	<u>61.5</u>	<u>66.9</u>	<u>44.8</u>
HITRA (ours)	**50.8**	**64.2**	**70.2**	**47.9**

Overall, HITRA achieves optimal or near-optimal accuracy among the evaluated methods across context scales. The diagnostics and ablations validate its coupled design: selective snapshot recall recovers evidence, hierarchical credit distinguishes decision failures, and listwise learning improves candidate selection, jointly strengthening long-context question answering.

6 Conclusion

HITRA combines selective snapshot recall, hierarchical credit assignment, and listwise trajectory learning for long-context agents. Recoverable memory preserves earlier hypotheses under a bounded active context; factorized feedback links intermediate decisions to task outcomes; and a separable listwise objective trains the shared policy to rank concurrent trajectories. Experiments across context scales and controlled diagnostics show gains in evidence recovery, credit localization, and trajectory selection. Future work includes learned snapshot retention and recall, richer credit for multimodal and tool-using agents, and adaptive rollout budgets across tasks and model scales.

AI USE STATEMENT

Generative AI tools were used for language polishing and to assist with retrieval and discovery of relevant literature. Their use focused on improving grammar, wording, and readability, and identifying potentially relevant studies. The authors reviewed all AI-assisted edits and verified the retrieved references against the original sources, and take full responsibility for the final manuscript.

REFERENCES

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *International Conference on Learning Representations (ICLR)*, 2026.

Wujiang Xu et al. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Bernal Jiménez Gutiérrez et al. From RAG to memory: Non-parametric continual learning for large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Jason Weston et al. StepWiser: Stepwise generative judges for wiser reasoning. *arXiv preprint arXiv:2508.19229*, 2025.

Yifei Zhou et al. SWEET-RL: Training multi-turn LLM agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

Lang Feng et al. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Pranav Putta et al. Agent Q: Advanced reasoning and learning for autonomous AI agents. *arXiv preprint arXiv:2408.07199*, 2024.

Hyungjoo Chae et al. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *International Conference on Learning Representations (ICLR)*, 2025.

Sikuan Yan et al. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Yue Yu et al. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020.

Aydar Bulatov, Yury Kuratov, and Mikhail S. Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, 2022, pp. 11079–11091.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12, 2024, pp. 157–173.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022, pp. 24824–24837.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023, pp. 8634–8652.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023, pp. 46534–46594.

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. In *Advances in Neural Information Processing Systems*, 2023a, pp. 11809–11822.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 62138–62160.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022, pp. 27730–27744.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

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

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 129–136.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: Rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*, 2023, pp. 10935–10950.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023, pp. 53728–53741.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2369–2380.

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. *arXiv preprint arXiv:2412.15115*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025, pp. 1279–1297.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024, pp. 62557–62583.

A Structured Candidate Interface

At each non-final recurrent step, the policy is instructed to emit the following fields:

```
<thinking>brief reasoning about the current state</thinking>
<plan>one concrete subgoal</plan>
<action>one valid action or retrieval decision</action>
<evidence>new grounded evidence or observation</evidence>
<uncertainty>unresolved claims or missing evidence</uncertainty>
<trajectory>a compact updated trajectory state</trajectory>
<expand>optional branch-expansion query</expand>
```

The trajectory and uncertainty fields are retained as recurrent state. If an expansion query is present, it is matched only against snapshots from the same rollout, and the top retrieved snapshot is inserted into the next prompt. At the final turn, the model receives the question, current trajectory state, and recalled evidence and produces the answer in a task-specific canonical format. The `thinking` field is a task-facing reasoning summary used for training and debugging; it is not treated as a faithful explanation of internal model computation.

B Training Procedure

For clarity, one global policy update consists of the following operations:

1. Sample a batch of problems and replicate each problem K times.
2. Initialize independent recurrent state and snapshot memory for every rollout.
3. For each context chunk, generate one turn per rollout, parse its fields, store the pre-update state, update recurrent state, and optionally recall a snapshot.
4. Generate a final answer from each completed trajectory.
5. Compute terminal task reward and local plan, action, format, and breadth-sensitive signals.
6. Recompute behavior-policy and frozen-reference token log-likelihoods for every retained turn.
7. Gather scalar scores by `(task_id, step_id)`; form component-wise local advantages, overall advantages, mixed policy advantages, and detached listwise weights.
8. Compute the global response-token count N_{tok} and intermediate-cohort count N_{coh}. Split turns into token-balanced microbatches and accumulate Eq. (6) using these fixed normalizers and the distributed reduction correction described in Appendix C.
9. Synchronize the updated actor with the rollout engine before collecting the next batch.

C Experimental Protocol and Implementation Details

Rollout records and dynamic batching. The behavior policy remains fixed while all replicas process a chunk sequence and emit final answers. The `rollout_index` maps every retained turn to its terminal outcome, `turn_type` implements the intermediate and final masks, and `(task_id, step_id)` identifies a synchronized cohort. Behavior and reference log-likelihoods, cohort statistics, A_i, ω_i^{rew}, and ω_i^{br} are attached to turn records before length-based shuffling or sharding. Dynamic token batching then groups turns by realized length to reduce padding.

Packing invariance requires detached cohort statistics and fixed global normalizers. Each microbatch contributes its token-loss sum divided by N_{tok} and its ranking sum divided by $N_{\mathrm{coh}} K \vartheta_\pi$, while each response score remains normalized by its own n_i. Unequal microbatch means are not averaged equally. When a distributed reducer averages across N_{wrk} workers, each local, globally normalized sum is multiplied by N_{wrk}; equivalently, the reducer may sum. These rules preserve the objective and its gradient under repacking.

Distributed execution. The trainable actor is sharded with fully sharded data parallelism, and a frozen reference policy supplies the KL target. Actor weights are synchronized with the SGLang rollout engine before each rollout batch and remain fixed across recurrent chunk steps. Cohort softmaxes are computed from gathered scalar scores and rewards; only detached per-turn records enter the actor workers.

Data construction and evaluation. HotpotQA contexts preserve their original paragraph order and provenance and are partitioned into fixed-token chunks without moving gold evidence across splits. Training uses the official training partition, whereas model selection and all reported analyses use disjoint development examples. Additional distractors are sampled only from the training partition to avoid evaluation leakage.

Primary task metrics are answer exact match and token-level F1. Supporting-fact F1 and joint answer–support F1 measure evidence grounding. Trajectory diagnostics include parse success, evidence precision, recall frequency, unique-evidence rate, and Spearman correlation between cohort scores and terminal returns. Systems measurements include generated tokens, wall-clock time, peak accelerator memory, and completed rollouts per second. Repeated training runs report means and standard deviations; fixed-checkpoint scaling, memory, and selector studies use their corresponding fixed evaluation sets. Paired bootstrap confidence intervals are reported when per-example predictions are available.

Baselines and controlled ablations. Baselines include the instruction-tuned backbone, supervised trajectory fine-tuning, outcome-only GRPO, and recurrent GRPO under matched rollout count, token budget, prompts, and snapshot memory. Controlled ablations remove component-wise local advantages, snapshot recall, breadth-sensitive shaping, reward ranking, or breadth ranking. A full-graph ListNet run on a memory-feasible subset verifies Eq. (4); varying the microbatch token budget tests packing invariance. Sweeps over K, chunk size, α, and the ranking coefficients separate gains from sampling, hierarchical credit, and listwise training.

D Reward Masks and Credit Assignment

Intermediate turns receive plan, action, candidate-format, breadth-sensitive, and propagated outcome signals. Final turns receive outcome and final-format signals but are masked out of the trajectory-ranking auxiliary loss. A rollout-index map links every recurrent turn to its terminal answer, while a turn-type mask distinguishes intermediate and final records. Synchronized-cohort identifiers combine the task identifier and recurrent step. They do not assert equality of the rollout-specific recurrent states.

E First-order Surrogate Derivation

Fix one intermediate cohort and suppress its conditioning notation. Let $u_k = \ell_k(\theta)/\vartheta_\pi$ and $u_k^0 = \ell_k^{\mathrm{old}}/\vartheta_\pi$ be the current and behavior-policy logits for candidate k. Collect them in vectors $\boldsymbol{u}$ and $\boldsymbol{u}^0$, and define

$$F(\boldsymbol{u}) = \log \sum_{k=1}^{K} \exp(u_k).$$

Let $\boldsymbol{p}_\theta$, $\boldsymbol{p}_{\mathrm{old}}$, and $\boldsymbol{p}_\gamma$ collect the corresponding cohort probabilities, where $\gamma \in \{\mathrm{rew}, \mathrm{br}\}$ and $\boldsymbol{p}_\gamma$ is fixed during optimization. Then

$$\mathcal{L}_\gamma(\boldsymbol{u}) = F(\boldsymbol{u}) - \boldsymbol{p}_\gamma^\top \boldsymbol{u},$$

and

$$\nabla_{\boldsymbol{u}} \mathcal{L}_\gamma(\boldsymbol{u}) = \boldsymbol{p}_\theta - \boldsymbol{p}_\gamma.$$

Substituting Eq. (3) into the surrogate yields

$$\widetilde{\mathcal{L}}_\gamma(\boldsymbol{u}) = (\boldsymbol{p}_{\mathrm{old}} - \boldsymbol{p}_\gamma)^\top (\boldsymbol{u} - \boldsymbol{u}^0).$$

The loss difference has the exact decomposition

$$\mathcal{L}_\gamma(\boldsymbol{u}) - \mathcal{L}_\gamma(\boldsymbol{u}^0) = \widetilde{\mathcal{L}}_\gamma(\boldsymbol{u}) + D_F(\boldsymbol{u}, \boldsymbol{u}^0),$$

where

$$D_F(\boldsymbol{u}, \boldsymbol{u}^0) = F(\boldsymbol{u}) - F(\boldsymbol{u}^0) - \boldsymbol{p}_{\mathrm{old}}^\top(\boldsymbol{u} - \boldsymbol{u}^0) = D_{\mathrm{KL}}(\boldsymbol{p}_{\mathrm{old}} \| \boldsymbol{p}_\theta).$$

The remainder is a KL between *cohort selection distributions*, distinct from the reference-policy token KL in Eq. (5). It is non-negative and has zero first derivative at $\boldsymbol{u} = \boldsymbol{u}^0$. The chain rule therefore proves Eq. (4) for either target. Equivalently, the parameter-gradient discrepancy away from the behavior policy is

$$\nabla_\theta \mathcal{L}_\gamma^{\mathcal{G}} - \nabla_\theta \widetilde{\mathcal{L}}_\gamma^{\mathcal{G}} = \frac{1}{\vartheta_\pi} \sum_{k=1}^{K} (p_\theta(k) - p_{\mathrm{old}}(k)) \nabla_\theta \ell_k(\theta).$$

This equality gives a precise local guarantee without claiming matching Hessians or global equivalence over multiple update epochs. When $K = 1$, both targets and the policy distribution equal one, so all ranking weights and ranking gradients vanish. When targets are uniform but actor scores are not, ranking still encourages a uniform cohort distribution; constant rewards do not by themselves imply a zero ranking gradient.

Let N_{mb} be the number of microbatches, and let the disjoint turn sets $\mathcal{I}_1, \ldots, \mathcal{I}_{N_{\mathrm{mb}}}$ have union $\mathcal{I}$. The batch ranking surrogate can be accumulated as

$$\overline{\mathcal{L}}_\gamma(\theta) = \sum_{\kappa=1}^{N_{\mathrm{mb}}} \frac{1}{N_{\mathrm{coh}} K \vartheta_\pi} \sum_{\substack{i \in \mathcal{I}_\kappa \\ \iota_i = 1}} \omega_i^\gamma [\ell_i^{\mathrm{old}} - \ell_i(\theta)].$$

Weights are computed from each turn's synchronized cohort before sharding; they are never recomputed within a microbatch. For gradients evaluated at the same θ, this identity is exact for the surrogate at any parameter value, up to numerical summation and stochastic forward-pass differences. Its equality to the full ListNet gradient is restricted to θ_{old}.