TRACEAHEAD: LEARNING NON-MYOPIC CONTROL FOR LLM SEARCH AGENTS FROM BRANCHED EXECUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM search agents often need to decide whether to keep searching and, if so, which direction to pursue, even though the value of a search action may only become apparent several steps later. Judging a search action only by its immediate effect can therefore overlook actions that enable valuable future searches.

We introduce **TraceAhead**, a lightweight value-based controller for non-myopic control of LLM search agents. TraceAhead models search control as a finite-horizon decision problem, using stopping as a zero-value reference and valuing each search action by its immediate net gain together with the downstream continuation it enables. It learns these history-dependent action values from branched training executions: starting from shared checkpoints, alternative actions are executed and regularized backward fitting propagates downstream value from successor histories to earlier decisions. At inference time, the learned values guide the agent's decision of whether to stop and, if continuing, which search direction to pursue, while only the selected action is executed along a single search path.

We evaluate TraceAhead on three multi-hop QA benchmarks with three LLM actors. At development-selected operating points with similar token use, TraceAhead achieves higher final-answer quality than three alternative search-control policies in all nine model–benchmark settings, with gains of up to 8.2 F1 points over the immediate-value policy. Additional analyses indicate that the learned continuation signal contains predictive information about future search opportunities. Together, these results show that training-time branches can provide supervision for history-dependent, non-myopic search decisions in LLM agents without requiring branch expansion at inference time.

1 INTRODUCTION

LLM search agents often need several retrieval steps to answer a complex question, with later queries depending on information uncovered earlier (Yao et al., 2023; Trivedi et al., 2023; Qian & Liu, 2025). This creates a basic difficulty for search control: a useful action need not immediately improve the current answer. For example, answering a question about the baggage allowance of the cheapest fare may first require identifying that fare. This query can leave the answer unchanged while revealing the entity needed for the next search; another query with the same immediate outcome may enable no useful continuation. Conversely, once sufficient evidence has been collected, further retrieval only adds cost. As illustrated in Figure 1(a), search actions with similar immediate effects can therefore have very different downstream value.

Recent work improves adaptive retrieval by learning when, what, or how much to search (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024; Qian & Liu, 2025; Java et al., 2026; Wu et al., 2026), while action-value methods evaluate language-model actions by their longer-term consequences (Bai et al., 2025; Dong et al., 2025). For a fixed search actor, however, two simpler forms of control leave an important gap. An immediate-value rule cannot credit a query whose benefit appears only after later searches, while a stage–action average continuation value assigns the same downstream correction to different histories. What matters instead is *history-specific continuation value*: whether an action is worth taking given the evidence already collected, the unresolved search state, and the remaining

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.

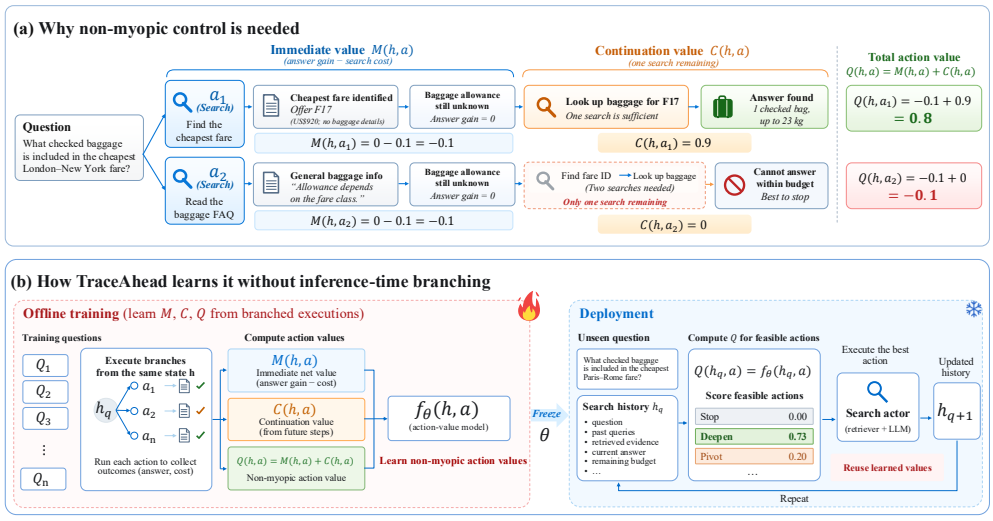


Figure 1: **Overview of TraceAhead.** **(a)** Equal immediate values $M(h,a)$ can yield different continuation $C(h,a)$ and total $Q(h,a) = M(h,a) + C(h,a)$. **(b)** Training branches fit frozen values that score feasible actions on unseen questions; only one path is executed, amortizing future-branch evaluation without online expansion. Values are illustrative.

budget. We therefore ask whether branches executed on training questions can teach a controller to recognize such downstream opportunities on unseen questions, without requiring alternative futures to be expanded at inference time.

We introduce **TraceAhead**, a lightweight value-based controller that moves branch expansion from deployment to training. At shared checkpoints on training questions, TraceAhead executes alternative feasible search actions while keeping the preceding history fixed. Each branch provides an immediate net reward—answer improvement minus search cost—and a successor history; regularized backward fitting propagates value from successor histories to earlier actions, producing stop-relative, history-dependent action-value estimates. At deployment, the LLM search actor and learned value heads are frozen. The controller scores stopping and the feasible search actions, executes only the selected action, and then repeats from the resulting history (Figure 1(b)). Thus, branches provide supervision about alternative futures during training, while inference follows a single search path. This distinguishes TraceAhead from inference-time tree search (Zhou et al., 2024) and from methods that use branched rollouts to optimize the underlying agent policy (Ji et al., 2026): here, branching is used to train a separate controller for a fixed actor.

We evaluate TraceAhead on MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), and 2WikiMultihopQA (Ho et al., 2020) with three LLM search actors, fitting a separate controller for each benchmark–actor setting. At development-selected operating points with closely matched mean logical-token use, TraceAhead achieves higher answer F1 than immediate-value, stage-average-continuation, and current-quality controls in all nine settings. On MuSiQue with DeepSeek v4.1 Flash, it improves F1 by 1.65–2.82 points over these controls at similar mean token use. We also report the surrounding quality–cost curves, where the relative advantage varies with operating point. Beyond final performance, cost-matched component comparisons test the contributions of continuation backup, search direction, and adaptive stopping, while independent suffix executions test whether the learned continuation increment predicts subsequently realized future return. The pooled suffix analysis shows a positive association, providing independent evidence that the learned signal contains information about downstream search opportunity.

Our contributions are:

- We develop TraceAhead, a stop-relative value controller for a fixed LLM search actor. Training branches and regularized backward fitting learn history-dependent downstream action value; deployment follows one path.

- Across three benchmarks and three LLM actors, TraceAhead achieves higher answer F1 than immediate-value, stage-average, and current-quality controls in all nine settings at selected, similar-token operating points.
- Cost-matched comparisons support continuation backup and adaptive stopping, with mixed effects for learned direction. Independent suffixes show a positive pooled association between the fitted continuation increment and future return.

2 Related Work

Adaptive retrieval and search control. ReAct and IRCoT interleave reasoning and retrieval (Yao et al., 2023; Trivedi et al., 2023). FLARE anticipates evidence needs, Self-RAG learns retrieval and critique, and Adaptive-RAG selects by question complexity (Jiang et al., 2023; Asai et al., 2024; Jeong et al., 2024). Search-R1 and FrugalRAG train search behavior and retrieval depth with reinforcement learning (Jin et al., 2025; Java et al., 2026). InForage rewards informative retrievals, HiPRAG provides process feedback, and DAS aligns search and answer decisions using counterfactual trajectories (Qian & Liu, 2025; Wu et al., 2026; Zhang et al., 2026). TraceAhead fixes the LLM actor and learns the downstream net value of stopping, deepening, or pivoting at each checkpoint.

Tree search and branched executions. Search trees expose alternative trajectories. LATS compares continuations during inference (Zhou et al., 2024). Tree-GRPO collects branched training rollouts for step-level policy optimization (Ji et al., 2026). TraceAhead uses shared-checkpoint branches to supervise a separate action-value controller; deployment scores actions without expanding alternative continuations.

Action-value learning for agents. Action-value methods estimate downstream effects. Digi-Q extracts a device-control policy from offline Q-values (Bai et al., 2025); LAC evaluates long-horizon LLM actions using future rollouts (Dong et al., 2025). TraceAhead combines immediate answer improvement minus query cost with a regularized downstream backup, then compares search actions against stopping while keeping the actor fixed.

3 Problem Setup

Search histories. Let $(x, y) \sim \mathcal{P}$ denote a question and reference answer, with the retrieval corpus included in the task. The fixed actor issues a shared first query before the controller acts, so h_1 is the first controllable checkpoint. After q queries, the observable history h_q contains the question, previous queries and evidence, the current answer $\widehat{y}(h_q)$, and route and verifier outputs, but not y or reference-based scores. A controller selects from $\overline{\mathcal{A}}_q(h_q) = \{\mathsf{Stop}\} \cup \mathcal{A}_q(h_q)$. A non-stopping action instructs the fixed actor–tool system ψ to produce $H_{q+1} \sim P_\psi(\cdot \mid h_q, a)$. Stopping returns the existing answer. In our interface, $\mathcal{A}_1 = \{\mathsf{Continue}\}$ and $\mathcal{A}_q = \{\mathsf{Deepen}, \mathsf{Pivot}\}$ for $1 < q < B$; the query cap forces stopping at B. Deepen pursues the current route; Pivot addresses another unresolved constraint. The actor turns each route instruction into a query.

Control objective. Let $F_y(h) = \mathcal{R}(\widehat{y}(h), y) \in [0, 1]$ denote answer quality, and let $c_q \geq 0$ denote the cost of a non-stopping transition. An observable-history policy $\pi \in \Pi_{\mathrm{obs}}$ induces a stopping time $\tau \leq B$. Given a mean budget $\overline{C}$, we consider the constrained objective and an associated fixed-price objective:

$$\max_{\pi \in \Pi_{\mathrm{obs}}} \mathbb{E}_\pi[F_Y(H_\tau)] \quad \text{s.t.} \quad \mathbb{E}_\pi\left[\sum_{q=1}^{\tau-1} c_q\right] \leq \overline{C}, \tag{1}$$

$$J_\lambda(\pi) = \mathbb{E}_\pi\left[F_Y(H_\tau) - \lambda \sum_{q=1}^{\tau-1} c_q\right], \quad \lambda \geq 0. \tag{2}$$

The fixed-price formulation follows constrained control (Altman, 1999). The shared first-query cost is omitted because it does not vary across the policies compared. Our fitted objective uses $c_q = 1$, so λ specifies a quality–query preference under the hard five-query cap. Experiments measure logical-token use separately from this training objective.

Values relative to stopping. For $a \neq \mathsf{Stop}$, define

$$Q^{\pi}_{\lambda,q}(h,a) = \mathbb{E}_{\pi}\left[F_Y(H_\tau) - F_Y(h) - \lambda \sum_{t=q}^{\tau-1} c_t \mid H_q = h, A_q = a\right], \quad Q^{\pi}_{\lambda,q}(h,\mathsf{Stop}) = 0. \tag{3}$$

The expectation covers answer uncertainty and stochastic execution. Subtracting $F_Y(h)$ makes stopping now the reference decision, with zero incremental value. Reference answers supply offline reward labels and are excluded from policy inputs.

4 METHOD

TraceAhead is a lightweight value-based controller that estimates the net utility of search actions, including their downstream continuations. The fixed actor supplies transitions, while the controller alone is fitted. Training branches execute alternative actions from the *same* history; backward fitting propagates their observed consequences to earlier decisions. At deployment, the fitted controller reads the observable history and executes only its selected action.

4.1 FROM IMMEDIATE PROGRESS TO CONTINUATION VALUE

For an executed transition, use the difference reward

$$\begin{aligned} r^{\lambda}_q &= F_y(h_{q+1}) - F_y(h_q) - \lambda c_q, \\ \sum_{q=1}^{\tau-1} r^{\lambda}_q &= F_y(h_\tau) - F_y(h_1) - \lambda \sum_{q=1}^{\tau-1} c_q. \end{aligned} \tag{4}$$

Because the initial checkpoint H_1 has the same distribution under all subsequent controllers, the second identity shows that maximizing cumulative difference reward is equivalent to maximizing J_λ: the two objectives differ by $\mathbb{E}[F_Y(H_1)]$. Appendix B.1 gives the finite-horizon proof, related to potential-based reward shaping (Ng et al., 1999). Suppressing λ in value notation, the Bellman recursion over full observable histories is

$$\begin{aligned} Q^*_q(h,a) &= \underbrace{\mathbb{E}[r^{\lambda}_q \mid h,a]}_{M_q(h,a)} + \underbrace{\mathbb{E}[V^*_{q+1}(H') \mid h,a]}_{C^*_q(h,a)}, \\ V^*_q(h) &= \max\{0, \max_{a \in \mathcal{A}_q(h)} Q^*_q(h,a)\}, \qquad V^*_B = 0. \end{aligned} \tag{5}$$

The immediate value M_q and continuation value C^*_q are different quantities. For example, identifying an intermediate entity may cost a query without changing the answer, but can make a subsequent query valuable. If $\max_a M_q(h,a) \leq 0$ while some $M_q(h,a) + C^*_q(h,a) > 0$, a one-step rule that favors Stop on ties stops although continuing is optimal. This decision-theoretic distinction is related to metareasoning (Hay et al., 2012; Callaway et al., 2018).

4.2 OFFLINE BRANCH SUPERVISION AND BACKWARD FITTING

Alternative actions are executed from common training checkpoints to obtain $\mathcal{D}_q = \{(h_i, a_i, h'_i, r^{\lambda}_i)\}_{i=1}^{n_q}$. Comparing branches at a shared checkpoint changes the action while holding the preceding evidence fixed. All branches of a question remain in the same data fold. For a fixed successor function v, function class $\mathcal{F}_q$, and regularizer Ω_q, define

$$\widehat{\mathcal{B}}_q v \in \arg\min_{f \in \mathcal{F}_q} \left\{ \sum_{i=1}^{n_q} [f(h_i,a_i) - r^{\lambda}_i - v(h'_i)]^2 + \alpha \Omega_q(f) \right\}. \tag{6}$$

Successor targets are held fixed during each regression, giving fitted action-value learning (Ernst et al., 2005; Munos & Szepesvári, 2008). Initialize $\widehat{V}^-_B = 0$ and, backwards in q, compute

$$\begin{aligned} \widehat{Q}_q &= \widehat{\mathcal{B}}_q \widehat{V}^-_{q+1}, \\ \widehat{V}^-_q(h) &= \max\{0, \max_a [\widehat{Q}_q(h,a) - \Gamma_q(h,a)]\}. \end{aligned} \tag{7}$$

The nonnegative Γ_q lowers the successor value used in earlier fitted targets, following the estimate–penalize–back-up organization of offline pessimistic control (Jin et al., 2021; Xie et al., 2021). At the last decision the target is only immediate reward. Moving backward, each earlier target adds the value of the best subsequent decision after its executed transition. This is how a useful later query can affect the score of the action that made it possible. With $B = 5$, full remaining-horizon fitting permits four non-stopping transitions from h_1 and is denoted H4; Appendix B.5 defines shorter receding horizons.

4.3 Ridge Estimation and the Backup Penalty

We instantiate $\widehat{Q}_q(h, a) = \phi_q(h)^\top \widehat{w}_{q,a}$ with separate stage/action heads and $\Omega_q(f) = \sum_{a \in \mathcal{A}_q} \|w_{q,a}\|_2^2$ in Equation (6). The seven heads contain 393 fitted mean coefficients. Features summarize public verifier outputs, answer/evidence overlap, and changes in retrieval; their exact definition is in Appendix D.2. They exclude gold scores and unexecuted descendants. With $z_i = \phi_q(h_i)$, $t_i = r_i^\lambda + \widehat{V}_{q+1}^-(h_i')$, and $\mathcal{I}_{q,a} = \{i : a_i = a\}$, ridge regression gives

$$\begin{aligned} \Lambda_{q,a} &= \alpha I + \sum_{i \in \mathcal{I}_{q,a}} z_i z_i^\top, \\ \widehat{w}_{q,a} &= \Lambda_{q,a}^{-1} \sum_{i \in \mathcal{I}_{q,a}} z_i t_i. \end{aligned} \tag{8}$$

Define $\mathcal{W}_{q,a}(\kappa) = \{w : \|w - \widehat{w}_{q,a}\|_{\Lambda_{q,a}} \le \kappa\}$. Its pointwise lower envelope is

$$\inf_{w \in \mathcal{W}_{q,a}(\kappa)} \phi_q(h)^\top w = \widehat{Q}_q(h, a) - \underbrace{\kappa \sqrt{\phi_q(h)^\top \Lambda_{q,a}^{-1} \phi_q(h)}}_{\Gamma_q(h,a)}. \tag{9}$$

The ridge geometry makes the penalty larger where the feature direction is less supported by the training design. We select κ on development data; Appendix B.3 derives the lower envelope.

4.4 Frozen Single-Path Control

The backup penalty enters successor targets during fitting. The evaluated deployment policy instead compares the fitted means with Stop's zero incremental value:

$$\begin{aligned} \widehat{\pi}_q(h) &= \operatorname*{arg\,max}_{a \in \{\mathsf{Stop}\} \cup \mathcal{A}_q(h)} s_q(h, a), \\ s_q(h, \mathsf{Stop}) &= 0, \qquad s_q(h, a) = \widehat{Q}_q(h, a) \quad (a \neq \mathsf{Stop}). \end{aligned} \tag{10}$$

Ties favor stopping; the cap forces stopping. Cost-curve operating points add a development-selected offset to non-Stop scores, changing stopping tendency while retaining the fitted query price. Each selected action is executed by the fixed actor, after which the controller reads the new history. Parameters remain frozen within and across questions. Deployment executes one path with its ordinary planning, answering, and verification calls.

The complete training and deployment sequence is summarized in Algorithm 1.

Algorithm 1 TraceAhead: execution-supervised continuation control

Require: Training questions and references; fixed actor–tool system ψ; $B, \lambda, \alpha, \kappa$; features ϕ

Collect common-checkpoint branches; form $\mathcal{D}_q$ using Equation (4)
$\widehat{V}_B^- \leftarrow 0$
for $q = B - 1, \dots, 1$ **do**
    Fit $\widehat{Q}_q \leftarrow \widehat{\mathcal{B}}_q \widehat{V}_{q+1}^-$; construct Γ_q by Equation (9)
    $\widehat{V}_q^-(h) \leftarrow \max\{0, \max_a[\widehat{Q}_q(h, a) - \Gamma_q(h, a)]\}$
end for
Freeze all heads; obtain shared first checkpoint h_1 for a new question
for $q = 1, \dots, B - 1$ **do**
    $a \leftarrow \widehat{\pi}_q(h_q)$; if $a = \mathsf{Stop}$, return $\widehat{y}(h_q)$
    Execute only a using ψ and obtain h_{q+1}
end for
return $\widehat{y}(h_B)$

The learned increment. To define a mechanism diagnostic, consider the Myopic policy, which replaces every target by r_i^λ with the same data, features and regularizer; it may still execute multiple queries. Define

$$\widehat{C}_q(h,a) = \widehat{Q}_q^{\text{full}}(h,a) - \widehat{Q}_q^{\text{myopic}}(h,a). \tag{11}$$

For a matched linear head this equals $\phi_q(h)^\top \Lambda_{q,a}^{-1} X_{q,a}^\top \mathbf{v}$, where $\mathbf{v}$ contains successor targets (Appendix B.4). The projected increment can be negative, and independent suffix executions test its predictive information.

5 Experiments

We compare complete policies across three benchmarks and three LLM actors at closely matched token costs, inspect representative quality–cost operating points, test component contributions with DeepSeek v4.1 Flash across benchmarks, and examine whether the learned continuation increment predicts future returns.

5.1 Evaluation Design

The MuSiQue evaluation uses task-local candidate-paragraph retrieval (Trivedi et al., 2022). A lexical retriever returns one paragraph per query, with a five-query cap. Questions come from the official train pool. The controller chooses a route; the actor writes the query. HotpotQA and 2WikiMultihopQA use their respective task adapters; each actor is held fixed within a comparison.

Five-fold task-level development selection chose $\alpha = 0.25$, $\kappa = 0.25$, and H4 at $\lambda = 0.025$ from 147 training trees for the MuSiQue/DeepSeek v4.1 Flash anchor. Each benchmark has a separate 400-question panel drawn by simple random sampling without replacement and shared across actors and policies; the MuSiQue panel comes from the official-train internal 2/3-hop pool. Evaluation question IDs are disjoint from those used to fit their corresponding controllers. The component-comparison table reuses these same evaluation panels under DeepSeek v4.1 Flash.

We score final answers with each benchmark's official answer evaluator. F1 is the mean per-question token-overlap F1, expressed as a percentage; exact match (EM) is the percentage of answers matching a reference under that evaluator's normalization. Tokens (k) is the mean, in thousands per question, of input plus output tokens over the deployed policy's logical model calls, including query planning, candidate answering, verification, and format repairs. Token counts are compared within an actor because model providers use different tokenizers. For complete-policy differences, pointwise paired bootstrap intervals resample question indices with policies and offsets frozen (Efron & Tibshirani, 1993); component-variant intervals preserve prespecified hop strata and recompute the cost interpolation. Appendix C.1 specifies both calculations.

Myopic. Myopic asks whether the *next query alone* pays for itself. It uses TraceAhead's branched transitions, public features, stage/action heads, and ridge regularizer, but fits immediate answer-F1 gain minus query cost. Stop scores zero; the best positive action is chosen. The decision is recomputed after each query, so Myopic may use all five queries. Development-selected offsets on non-Stop scores set its cost points.

SA Mean. SA Mean asks whether a *typical* later payoff for each stage/action is enough. It fits immediate-action heads on Myopic's features and transitions, with its own development-selected ridge precision, then scores $s_q(h,a) = Q_q^{\text{SA,1step}}(h,a) + \beta b_{q,a} + \delta$, where $b_{q,a} = n_{q,a}^{-1} \sum_{i \in \mathcal{I}_{q,a}} \widehat{V}_{q+1}^-(h_i')$ averages risk-adjusted successor values ($b_{B-1,a} = 0$). Only the immediate term varies by history. Development trajectories select $\beta \in \{0, 0.25, \ldots, 2\}$ and δ; Stop scores zero.

Quality Threshold. Quality Threshold asks whether the *current answer* is good enough. It regresses current-answer F1 from the same public features and stops above a development-selected threshold. Its continuation route is chosen on development trajectories from either the verifier recommendation or learned immediate-action scores. It can choose where to search, but does not fit downstream continuation value. All three controls share the five-query cap and select settings only on development trajectories.

5.2 Complete-Policy Results Across Benchmarks and Actors

Table 1: **Complete-policy results across benchmarks and LLM actors.** Each benchmark has 400 questions shared across actors and policies; controllers are fitted separately by setting. Parentheses on control rows give TraceAhead minus control in F1 and EM points, and relative logical-token savings (positive means fewer tokens for TraceAhead). Metrics are rounded to one decimal; bold marks the best unrounded value per setting and metric. Pointwise paired 95% F1 intervals are in Appendix Table 3.

Benchmark	Policy	F1 ↑	EM ↑	Tokens (k) ↓
GPT-6 Luna				
MuSiQue	Myopic	49.6 (+5.9)	38.3 (+4.8)	**15.3** (-1.7%)
	SA Mean	51.9 (+3.5)	39.8 (+3.3)	15.5 (-0.4%)
	Quality Threshold	53.8 (+1.7)	41.5 (+1.5)	15.8 (+1.9%)
	TraceAhead	**55.4**	**43.0**	15.5
HotpotQA	Myopic	76.4 (+0.9)	61.0 (+0.8)	14.8 (+1.1%)
	SA Mean	75.8 (+1.6)	60.5 (+1.3)	**14.4** (-2.0%)
	Quality Threshold	74.7 (+2.7)	58.8 (+3.0)	15.0 (+2.0%)
	TraceAhead	**77.3**	**61.8**	14.7
2Wiki	Myopic	66.3 (+1.6)	58.8 (+1.8)	16.5 (-1.3%)
	SA Mean	62.5 (+5.4)	55.8 (+4.8)	**16.4** (-1.9%)
	Quality Threshold	65.2 (+2.8)	58.0 (+2.5)	17.1 (+1.9%)
	TraceAhead	**67.9**	**60.5**	16.7
DeepSeek v4.1 Flash				
MuSiQue	Myopic	58.5 (+2.1)	47.0 (+1.5)	14.8 (+0.3%)
	SA Mean	59.0 (+1.7)	47.0 (+1.5)	15.0 (+1.8%)
	Quality Threshold	57.8 (+2.8)	46.5 (+2.0)	**14.4** (-1.9%)
	TraceAhead	**60.6**	**48.5**	14.7
HotpotQA	Myopic	79.6 (+1.1)	64.8 (+1.0)	13.8 (-1.1%)
	SA Mean	79.6 (+1.1)	64.8 (+1.0)	14.2 (+1.9%)
	Quality Threshold	79.5 (+1.2)	64.0 (+1.8)	**13.7** (-1.9%)
	TraceAhead	**80.7**	**65.8**	13.9
2Wiki	Myopic	82.0 (+1.6)	74.3 (+1.5)	**14.9** (-1.7%)
	SA Mean	80.2 (+3.4)	72.0 (+3.8)	15.4 (+1.8%)
	Quality Threshold	81.5 (+2.1)	73.3 (+2.5)	14.9 (-1.4%)
	TraceAhead	**83.7**	**75.8**	15.1
GLM-5.3 Flash				
MuSiQue	Myopic	50.0 (+8.2)	38.3 (+7.5)	13.4 (-1.4%)
	SA Mean	52.6 (+5.6)	40.5 (+5.3)	**13.4** (-1.5%)
	Quality Threshold	55.7 (+2.5)	43.5 (+2.3)	13.5 (-0.5%)
	TraceAhead	**58.2**	**45.8**	13.6
HotpotQA	Myopic	80.6 (+0.4)	66.0 (+0.3)	14.2 (+0.6%)
	SA Mean	79.7 (+1.4)	63.8 (+2.5)	13.8 (-1.9%)
	Quality Threshold	78.1 (+2.9)	63.0 (+3.3)	**13.8** (-2.0%)
	TraceAhead	**81.0**	**66.3**	14.1
2Wiki	Myopic	80.1 (+1.5)	70.3 (+1.5)	15.4 (+1.0%)
	SA Mean	74.7 (+6.9)	63.5 (+8.3)	**15.0** (-2.0%)
	Quality Threshold	79.2 (+2.4)	68.3 (+3.5)	15.6 (+2.0%)
	TraceAhead	**81.6**	**71.8**	15.3

Table 1 tests the complete policies at development-selected, closely matched-token operating points. TraceAhead has higher F1 than each control in all nine benchmark–actor settings, with less than 2% difference in mean logical tokens per comparison. Against Myopic, gains range from 2.1 to 8.2 F1

points on MuSiQue, 0.4 to 1.1 on HotpotQA, and 1.5 to 1.6 on 2WikiMultihopQA; the magnitude varies substantially by benchmark. All 27 paired F1 intervals in Appendix Table 3 exclude zero. We use the MuSiQue/DeepSeek v4.1 Flash setting as the anchor for the subsequent cost and suffix analyses.

5.3 Quality–Cost Behavior Across Settings

Figure 2 places the selected-point results in their token-cost context. Panel (a) is the MuSiQue/DeepSeek v4.1 Flash anchor; (b) changes the benchmark with DeepSeek fixed, and (c) changes the actor with MuSiQue fixed. The middle points correspond to Table 1. TraceAhead leads at these selected points, while the curves converge or cross at other token costs. The remaining curves appear in Appendix A.2.

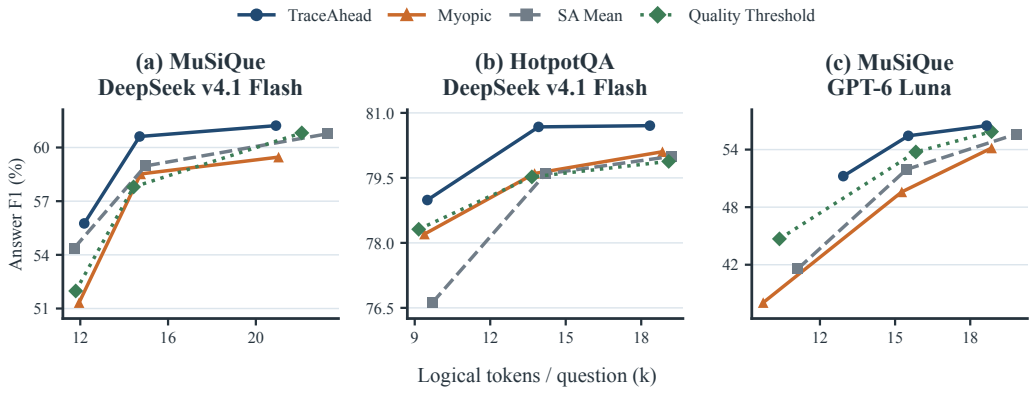


Figure 2: **Quality–cost behavior across settings.** (a) MuSiQue/DeepSeek v4.1 Flash; (b) HotpotQA/DeepSeek v4.1 Flash; (c) MuSiQue/GPT-6 Luna. Each panel shows four frozen policies at three executed points on 400 questions. Axes have panel-specific ranges; model tokenizers differ. Lines connect observations only.

5.4 Cost-Matched Component Comparisons

Table 2: **Cost-matched component comparisons on 400 questions per benchmark.** Cells report EM/F1 (%) with DeepSeek v4.1 Flash. Full TraceAhead is measured. The Myopic column reuses the policy in Table 1; its values and those of the two component variants are interpolated to Full's measured mean logical-token cost. Paired F1 intervals for the two component variants are in Appendix Table 4.

Benchmark	Myopic (Table 1)	w/o learned direction	w/o adaptive stopping	TraceAhead (Full)
	EM/F1 (%) ↑	EM/F1 (%) ↑	EM/F1 (%) ↑	EM/F1 (%) ↑
HotpotQA	64.78/79.62	65.18/80.28	64.16/78.63	**65.75/80.68**
2WikiMultihopQA	74.28/82.09	74.40/82.31	72.56/80.57	**75.75/83.67**
MuSiQue	46.89/58.40	47.44/58.98	44.44/55.79	**48.50/60.62**

For each benchmark, Full TraceAhead was rerun on the shared 400-question panel from the first-query checkpoints; its measured mean logical-token cost C^* is the target. Myopic is the Table 1 policy fitted to immediate net reward, without continuation backup. The two other variants use the same questions, checkpoints, actor, training data, remaining rules, and five-query cap. They replace the continuation route with development-fixed Pivot while retaining Full's stop rule (w/o learned direction), or fix the query count while retaining learned direction (w/o adaptive stopping). Each comparator was measured at multiple cost points; Table 2 interpolates mean EM and F1 between the adjacent points bracketing C^*.

At the common mean cost, Full TraceAhead has the highest EM and F1 in all three benchmark rows. Its F1 advantage over cost-matched Myopic is 1.06, 1.58, and 2.22 points on HotpotQA, 2WikiMultihopQA, and MuSiQue, respectively. The paired intervals in Appendix Table 4 exclude zero for adaptive stopping on all three benchmarks and for learned direction on 2WikiMultihopQA and MuSiQue; the HotpotQA learned-direction interval includes zero.

5.5 Does the Increment Predict Future Opportunity?

In the branched training records, 189 of 1,029 eligible actions have nonpositive immediate net gain yet a positive continuation among sampled descendants (Appendix Figure 7). To test whether the learned increment predicts later opportunity, we force actions from frozen MuSiQue/DeepSeek v4.1 Flash checkpoints, follow the original TraceAhead suffix policy, and measure the return *after* the forced action:

$$G_{\text{future}} = [F_y(h_\tau) - F_y(h_q) - \lambda(\tau - q)] - [F_y(h_{q+1}) - F_y(h_q) - \lambda]\,. \tag{12}$$

We rank the frozen predictions from Equation (11) within each q1/q2 stage/action stratum (Figure 3). The equal-cell pooled Q5–Q1 future-net contrast is 8.9 [2.1, 16.8] F1 points (95% question-cluster bootstrap interval). In the heatmap, pooled centered future return is negative in Q1–Q3 and positive in Q4–Q5. The q2–Deepen Q5–Q1 interval excludes zero, while the q1–Continue and q2–Pivot intervals include zero. Thus the fitted increment predicts future return in the pooled analysis, with the clearest stratum result for q2–Deepen.

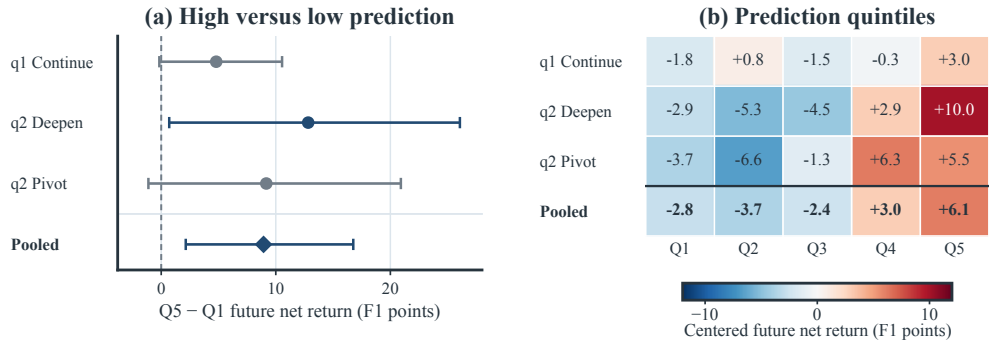


Figure 3: **Predicted continuation and independent future return on MuSiQue/DeepSeek v4.1 Flash.** (a) Observed Q5–Q1 future-net return and pointwise 95% question-cluster bootstrap intervals. (b) Observed means for all five prediction quintiles, centered within each stage/action stratum. Pooled results weight strata equally.

6 Conclusion

Effective LLM search control requires accounting for not only an action's immediate effect, but also the downstream opportunities it enables. TraceAhead learns this history-dependent value from alternative actions executed at shared training checkpoints, backing up downstream net value into action scores that jointly control stopping and search direction. Once fitted, the controller follows a single search path without expanding alternative futures online.

Across three multi-hop QA benchmarks and three LLM actors, separately fitted TraceAhead controllers improve final-answer F1 over three alternative search-control policies at development-selected operating points with similar token use. Cost-matched component comparisons support the roles of continuation backup and adaptive stopping, with more mixed evidence for learned search direction. Independent suffix executions further show a positive pooled association between the fitted continuation increment and later return. Together, these results support training-time branching as a source of reusable supervision for history-dependent, non-myopic control of LLM search agents.

AI USE STATEMENT

An AI assistant supported manuscript organization and revision, mathematical checks, reference formatting, and LaTeX and figure preparation.

REFERENCES

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999. URL https://www.routledge.com/Constrained-Markov-Decision-Processes/Altman/p/book/9780849303821.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.

Hao Bai, Yifei Zhou, Erran Li, Sergey Levine, and Aviral Kumar. Digi-Q: Learning VLM q-value functions for training device-control agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/519abe71ee55aac4fe821bbd731c6645-Abstract-Conference.html.

Frederick Callaway, Sayan Gul, Paul M. Krueger, Thomas L. Griffiths, and Falk Lieder. Learning to select computations. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018. URL https://www.auai.org/uai2018/proceedings/papers/269.pdf.

Heng Dong, Kefei Duan, and Chongjie Zhang. Enhancing decision-making of large language models via actor-critic. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13984–14020, 2025. URL https://proceedings.mlr.press/v267/dong25c.html.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, 1993.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(18):503–556, 2005. URL https://www.jmlr.org/papers/v6/ernst05a.html.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2012. URL https://www.auai.org/uai2012/papers/123.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. International Committee on Computational Linguistics, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Abhinav Java, Srivathsan Koundinyan, Nagarajan Natarajan, and Amit Sharma. FrugalRAG: Less is more in RL finetuning for multi-hop question answering. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ede6f43d254731152970009c172d5561-Abstract-Conference.html.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.21240.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Ö. Arık, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5084–5096. PMLR, 2021. URL https://proceedings.mlr.press/v139/jin21e.html.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(27):815–857, 2008. URL https://www.jmlr.org/papers/v9/munos08a.html.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999. URL https://people.eecs.berkeley.edu/~russell/papers/icml99-shaping.pdf.

Hongjin Qian and Zheng Liu. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/e4724af0e2a0d52ce5a0a4e084b87f59-Abstract-Conference.html.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

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/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.acl-long.557/.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. HiPRAG: Hierarchical process rewards for efficient agentic retrieval augmented generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Gt4v9WBPzm.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6683–6694, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/34f98c7c5d7063181da890ea8d25265a-Abstract.html.

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*, pp. 2369–2380. Association for Computational Linguistics, 2018. URL https://aclanthology.org/D18-1259/.

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*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Wenlin Zhang, Kuicai Dong, Junyi Li, Yingyi Zhang, Xiaopeng Li, Pengyue Jia, Yi Wen, Derong Xu, Maolin Wang, Yichao Wang, Yong Liu, and Xiangyu Zhao. To search or not to search: Aligning the decision boundary of deep search agents via causal intervention. In *Proceedings of the ACM Web Conference 2026*. Association for Computing Machinery, 2026. doi: 10.1145/3774904.3792235. URL https://doi.org/10.1145/3774904.3792235.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62138–62160. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24r.html.

A Additional Results

A.1 Paired F1 Intervals

Table 3: Pointwise paired-question 95% bootstrap intervals for the F1 differences in Table 1. Each cell compares TraceAhead with the named control on the same 400 questions.

Control	MuSiQue	HotpotQA	2Wiki
GPT-6 Luna			
Myopic	[3.3, 8.5]	[0.1, 1.9]	[0.3, 3.2]
SA Mean	[1.3, 5.8]	[0.3, 3.1]	[2.8, 8.3]
Quality Threshold	[0.2, 3.3]	[0.9, 4.5]	[0.4, 5.2]
DeepSeek v4.1 Flash			
Myopic	[0.9, 3.6]	[0.3, 2.1]	[0.6, 2.9]
SA Mean	[0.6, 3.0]	[0.3, 2.1]	[1.4, 5.7]
Quality Threshold	[1.2, 4.6]	[0.2, 2.3]	[0.6, 3.9]
GLM-5.3 Flash			
Myopic	[5.4, 11.1]	[0.1, 0.9]	[0.4, 2.8]
SA Mean	[3.1, 8.2]	[0.4, 2.5]	[4.3, 9.7]
Quality Threshold	[0.8, 4.3]	[1.1, 4.8]	[0.3, 4.5]

Table 4: **Paired F1 intervals for component variants.** Full minus each variant at Full's mean logical-token cost, in F1 points; parentheses give pointwise stratified paired-question 95% bootstrap intervals.

Variant	**HotpotQA**	**2WikiMultihopQA**	**MuSiQue**
w/o learned direction	$+0.402$ $(-0.587, 1.467)$	$+1.355$ $(0.289, 2.603)$	$+1.632$ $(0.626, 2.852)$
w/o adaptive stopping	$+2.044$ $(0.967, 3.243)$	$+3.097$ $(1.234, 4.944)$	$+4.825$ $(2.167, 7.494)$

A.2 Remaining Quality–Cost Curves

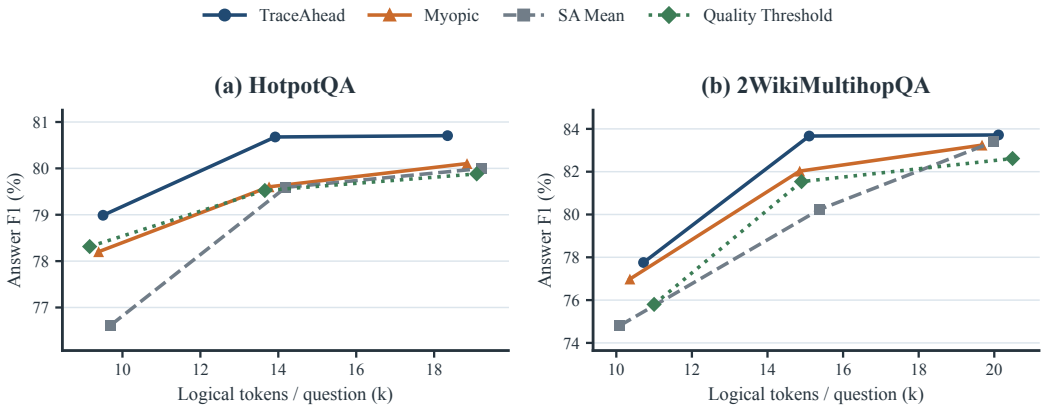


Figure 4: **Benchmark change with the DeepSeek v4.1 Flash actor fixed.** Answer F1 (%) against mean logical tokens per question (k) for four frozen policies at three measured points, on 400 questions per benchmark. MuSiQue/DeepSeek v4.1 Flash is in Figure 2; lines connect measured points.

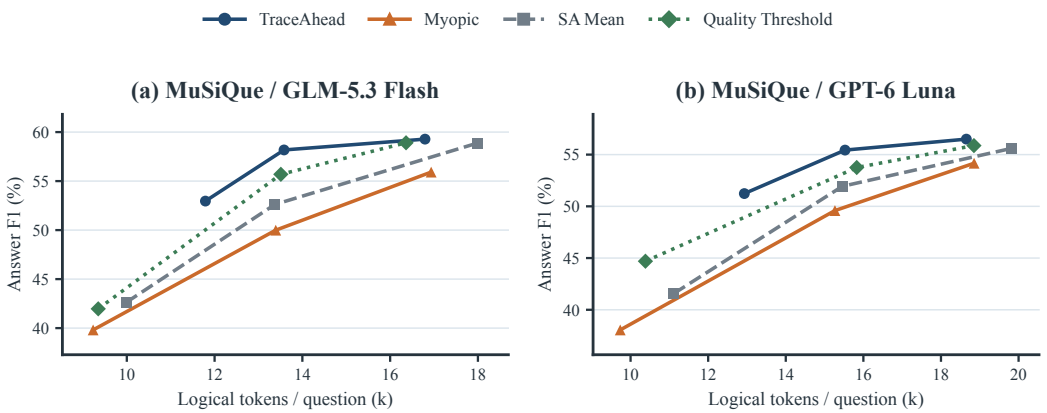


Figure 5: **Actor change on the same MuSiQue questions.** Answer F1 (%) against mean logical tokens per question (k) on the common 400-question panel. The DeepSeek v4.1 Flash anchor is in Figure 2.

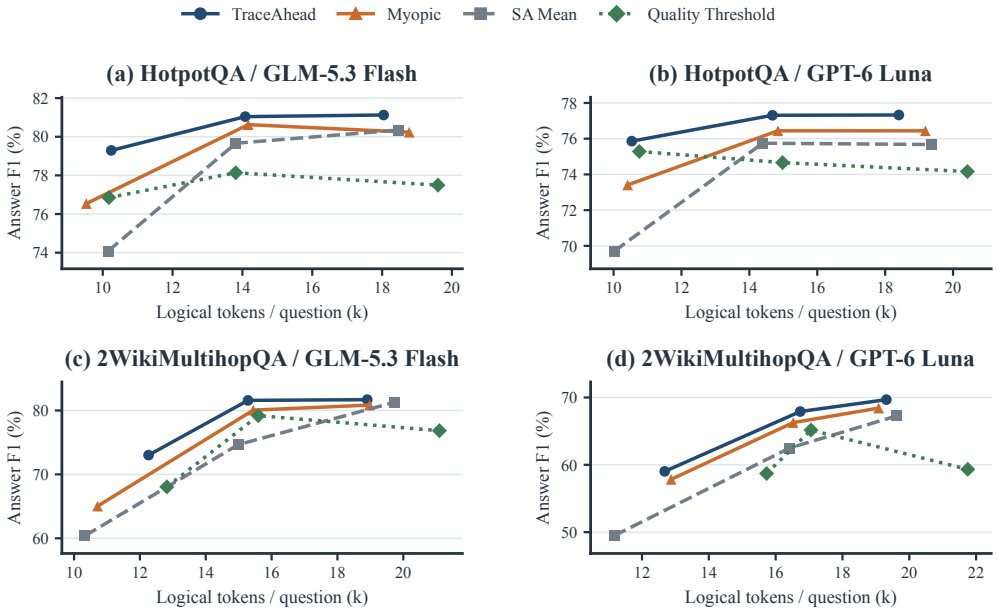


Figure 6: **Remaining benchmark–actor combinations.** Top row: HotpotQA with GLM-5.3 Flash and GPT-6 Luna; bottom row: 2WikiMultihopQA with the same actors. Each panel plots measured answer F1 against mean logical tokens on that benchmark's randomly sampled 400 questions. The middle operating points correspond to Table 1; lines connect measured points only.

A.3 Delayed Payoffs in Branched Executions

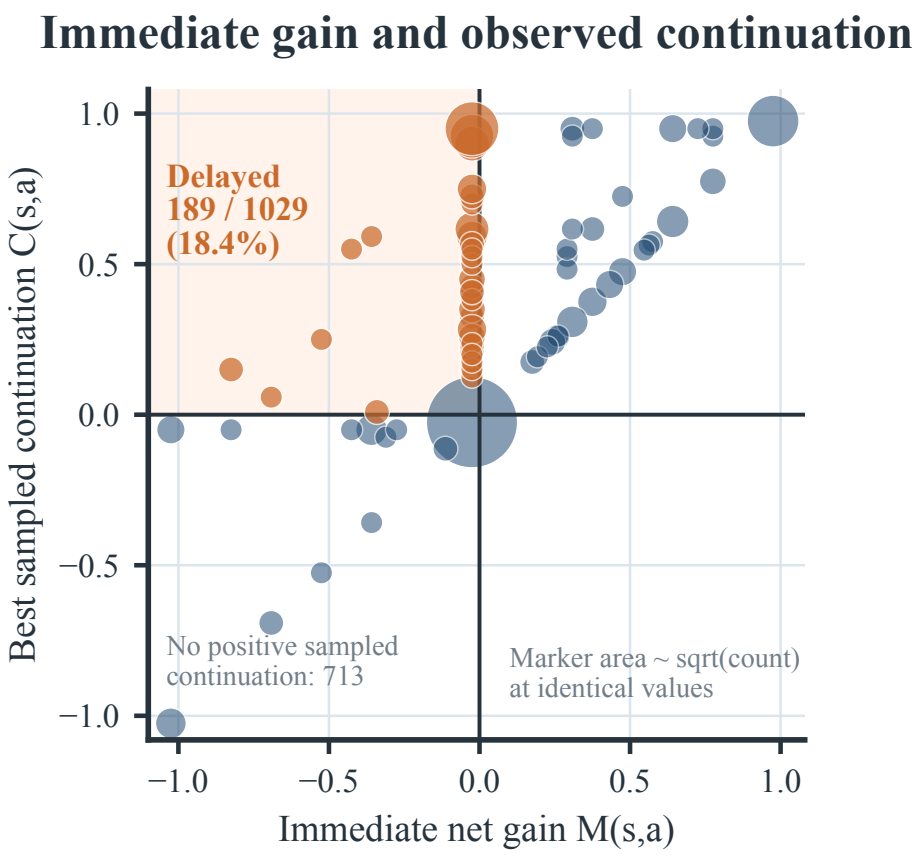


Figure 7: **Immediate gain and observed continuation in training branches.** The 1,029 q1–q3 action instances come from branched executions on 147 MuSiQue/DeepSeek v4.1 Flash training and development questions. The horizontal axis is the realized change in answer F1 after the action minus its query cost (Equation 4); the vertical axis is the best subsequent net return observed among sampled descendants. Each marker represents a distinct coordinate pair; larger markers denote more action instances at that pair. The orange upper-left region contains 189 actions (18.4%): their immediate net gain is nonpositive, while at least one sampled continuation has positive net return.

B MATHEMATICAL DETAILS

These derivations specify the estimator and the reference point used by the controller. Throughout, the query horizon is finite, transition costs are nonnegative and integrable, and stopping returns the candidate answer already present in the observable history.

B.1 EQUIVALENCE OF THE INCREMENTAL AND TERMINAL OBJECTIVES

Proposition 1 (Stop-relative reward equivalence)**.** *Assume that the distribution of the first-query checkpoint H_1 is independent of the subsequent controller. For every observable-history policy π with stopping time $\tau \le B$,*

$$\mathbb{E}_\pi\left[\sum_{q=1}^{\tau-1} r_q^\lambda\right] = J_\lambda(\pi) - \mathbb{E}[F_Y(H_1)]. \tag{13}$$

Consequently, both objectives have the same maximizers over Π_{obs}.

Proof. Fix a task, an execution realization, and its stopping time. Summing the reward definition gives

$$\sum_{q=1}^{\tau-1}\left[F_y(h_{q+1}) - F_y(h_q) - \lambda c_q\right] = F_y(h_\tau) - F_y(h_1) - \lambda\sum_{q=1}^{\tau-1} c_q,$$

including the empty sum when $\tau = 1$. Taking expectations proves Equation (13). The subtracted expectation is constant across the policies being compared. □

The difference-reward construction is a finite-horizon specialization of the reward-transformation perspective of Ng et al. (1999).

B.2 BELLMAN RECURSION ON OBSERVABLE HISTORIES

Conditioning on a non-stopping first action and adding and subtracting $F_Y(H_{q+1})$ in Equation (3) gives

$$Q_q^\pi(h,a) = \mathbb{E}\left[r_q^\lambda + V_{q+1}^\pi(H_{q+1}) \mid h, a\right], \qquad V_q^\pi(h) = \mathbb{E}_{a\sim\pi_q(\cdot|h)}[Q_q^\pi(h,a)]. \tag{14}$$

The tower property applies because the successor history contains the earlier observable history and action. Maximizing over feasible future observable-history policies gives Equation (5); finite action sets and a finite horizon permit backward maximization. Stopping makes $V_q^*(h) \ge 0$, although a particular non-stopping action can have negative value. Therefore the algorithm takes a maximum with zero at the state-value level; it does not clip each non-stopping action to be nonnegative. This distinction also preserves the stated stop-first tie rule.

B.3 ELLIPSOIDAL LOWER ENVELOPE

Lemma 1 (Pointwise lower envelope)**.** *For a positive-definite matrix Λ, a center $\widehat{w}$, a vector z, and radius $\kappa \ge 0$,*

$$\inf_{(w-\widehat{w})^\top\Lambda(w-\widehat{w})\le\kappa^2} z^\top w = z^\top\widehat{w} - \kappa\sqrt{z^\top\Lambda^{-1}z}. \tag{15}$$

Proof. Write $u = \Lambda^{1/2}(w - \widehat{w})$. The constraint becomes $\|u\|_2 \le \kappa$, and the variable part of the objective is $(\Lambda^{-1/2}z)^\top u$. By Cauchy–Schwarz its minimum is $-\kappa\|\Lambda^{-1/2}z\|_2$. For $z \ne 0$, equality is attained at $u = -\kappa\Lambda^{-1/2}z/\|\Lambda^{-1/2}z\|_2$. The case $z = 0$ is immediate. □

For the ridge estimator, $\alpha > 0$ makes $\Lambda_{q,a}$ positive definite, so the lemma gives the penalty in Equation (9).

B.4 Matched Linear Decomposition

Fix a stage/action design matrix X and its immediate reward vector $\mathbf{r}$. Let $\mathbf{v}$ contain the fitted successor targets used by the full controller, and let $\Lambda = \alpha I + X^\top X$. Fitting both targets with the same design and regularizer yields

$$\widehat{w}^{\text{full}} = \Lambda^{-1} X^\top (\mathbf{r} + \mathbf{v}), \qquad \widehat{w}^{\text{myopic}} = \Lambda^{-1} X^\top \mathbf{r}. \tag{16}$$

Subtracting gives

$$\widehat{C}_q(h, a) = \phi_q(h)^\top \Lambda^{-1} X^\top \mathbf{v}. \tag{17}$$

This decomposition uses the same design and regularizer for the full and Myopic heads.

For an independently executed suffix that first takes a and thereafter follows a fixed policy π_c, the matching observable outcome is

$$\begin{aligned} G_{\text{total}} &= F_y(h_\tau) - F_y(h_q) - \lambda \sum_{t=q}^{\tau-1} c_t, \\ G_{\text{future}} &= G_{\text{total}} - r_q^\lambda. \end{aligned} \tag{18}$$

Averaging independent suffix realizations estimates future returns under the fixed continuation policy π_c. Their relationship with $\widehat{C}_q$ measures the increment's predictive information.

B.5 Finite Lookahead

For maximum lookahead H, define fits indexed by physical query stage q and remaining fitted depth $d \le \min(H, B - q)$. Depth one uses $t_i^{(1)} = r_i^\lambda$. For $d > 1$,

$$t_i^{(d)} = r_i^\lambda + \max\left\{0, \max_{a'}\left[\widehat{Q}_{q+1}^{(d-1)}(h_i', a') - \Gamma_{q+1}^{(d-1)}(h_i', a')\right]\right\}. \tag{19}$$

At stage q, deploy the head with depth $\min(H, B - q)$. For truncated horizons, this depth is reset at the next actual checkpoint, giving receding-horizon control. Thus a depth-one Myopic head may execute a long trajectory despite using only one-step targets. With $B = 5$, H4 reaches the terminal cap from the first controlled checkpoint.

C Evaluation Details

C.1 Cost Matching

In the component comparisons, Full TraceAhead's measured mean logical-token cost C^* is the target. Each comparator was run at multiple costs. Table 2 interpolates panel-mean EM and F1 between the adjacent points $(\bar{C}_L, \bar{M}_L)$ and $(\bar{C}_U, \bar{M}_U)$ bracketing C^*:

$$\begin{aligned} w &= \frac{C^* - \bar{C}_L}{\bar{C}_U - \bar{C}_L}, \qquad & \widehat{\bar{M}}_{\text{comp}}(C^*) &= (1 - w)\bar{M}_L + w\bar{M}_U, \\ \Delta_{\text{F1}} &= \bar{F}_{\text{Full}} - \widehat{\bar{F}}_{\text{comp}}(C^*). \end{aligned} \tag{20}$$

The same bracketing points and weight are used for EM and F1. Full is measured; comparator values at C^* are interpolated.

C.2 Paired Bootstrap Intervals

For complete-policy comparisons, each replicate resamples 400 questions with replacement and averages their paired F1 differences. The fitted policies and operating offsets remain fixed. The 2.5th and 97.5th percentiles form the pointwise 95% interval.

For component-variant intervals, each replicate resamples questions within the prespecified hop strata, preserving pairing across Full and every comparator cost point. It recomputes mean costs and F1, the bracketing points, and the interpolation in Equation (20); percentile endpoints give the 95% interval.

C.3 Independent Suffixes

At most one checkpoint per question is selected by prespecified hash order within balanced stage and TraceAhead/Myopic decision strata. Each feasible non-Stop action is forced, after which the frozen zero-offset TraceAhead controls the remaining suffix. Three independent executions are averaged per checkpoint/action. Predictions use Equation (11), and future returns use Equation (12). Ranks and high/low contrasts are computed within stage/action strata; bootstrap resampling clusters all suffixes by question.

D Implementation and Training Details

D.1 Search Interface

The actor, retriever, answer generator, and verifier are fixed within each policy comparison. The actors are DeepSeek v4.1 Flash, GLM-5.3 Flash, and GPT-6 Luna.

Each MuSiQue question has 20 candidate paragraphs containing evidence and distractors. BM25 (Robertson & Zaragoza, 2009) returns one paragraph per query. The actor sees retrieved paragraphs; the controller does not receive reference answers, support labels, or answer scores. HotpotQA and 2Wiki use their task-provided paragraph collections and answer evaluators.

A checkpoint is obtained after retrieval, candidate answering, and verification. The first controlled checkpoint offers Stop or Continue. Continue generates the next search route; subsequent checkpoints offer Stop, Deepen, or Pivot. Deepen pursues the current route. Pivot asks the actor to pursue an alternative unresolved constraint using the current route and verifier output. The actor supplies actual query strings. The last allowed retrieval is followed by an answer, and the fifth-query cap forces stopping. Stopping returns the current candidate.

Offline training executes all feasible branches to the five-query cap. A complete tree has 16 retrieval nodes and 15 controlled transitions. Deployment follows one chosen route. For the MuSiQue/DeepSeek v4.1 Flash training set, collecting 147 complete trees used 5,880 logical model calls and 15.84M logical tokens.

D.2 Public Features

Features use only the observable history: verifier outputs, the current answer, query and retrieved passage, retrieval score, and changes since the previous checkpoint. They summarize answer and evidence overlap, verifier uncertainty, and retrieval novelty. Interactions with the current verifier state give 33 features at the first controlled checkpoint and 60 thereafter. Seven separate stage/action heads contain 393 mean coefficients in total. No feature uses a reference answer, gold score, or unexecuted branch.

D.3 Fitting and Operating Policies

The training price is $\lambda = 0.025$ per additional query, with working observation-noise variance one. The MuSiQue development grid uses ridge precision $\alpha \in \{0.25, 1, 4, 16\}$ and $\kappa \in \{0, 0.25, 0.5, 1\}$ for depths $H \in \{2, 3, 4\}$; depth one has no future penalty. Five-fold task-level validation selects H4, $\alpha = 0.25$, and $\kappa = 0.25$. Myopic shares the selected regularizer. The final heads are fitted on all 147 complete training trees.

The same H4 learning design is used for benchmark-specific refits. At each cost operating point, a fixed offset δ_j is added to every non-Stop score, while Stop remains zero. It is selected on development trajectories, then held constant throughout the question.