Preventing Error Cascades in Long-Horizon Multimodal Agents with Edge-Reliability Graph Memory

Saman Forouzandeh*
School of Engineering, Royal Melbourne Institute of Technology University
Melbourne, Australia

Wei Peng
School of Engineering, Royal Melbourne Institute of Technology University
Melbourne, Australia

Xinghuo Yu
School of Engineering, Royal Melbourne Institute of Technology University
Melbourne, Australia

Mahdi Jalili
School of Engineering, Royal Melbourne Institute of Technology University
Melbourne, Australia

Abstract

Long-horizon tool-augmented agents suffer sharp degradation as trajectories grow: small tool errors are stored, repeatedly reused, and amplified into cascading failures. Existing approaches rely on post-hoc verification, item-level memory scoring, or executive context management, but do not directly control how unreliable evidence propagates through reuse. We propose **EPOCH** (Edge-Pathway Outcome-supervised Cascade-Halting memory), which mitigates cascades by learning an outcome-correlated, *policy-conditioned* utility on reuse pathways connecting observations over time, instantiated as a sparse memory graph with quality-weighted edges, a Temporal Graph Transformer over edge utilities, and a learned mid-context correction. **We show that quality-weighted and uniform retrieval are asymptotically separated in horizon length** whenever a single suppression constant is positive: the ratio of cumulative cascade error then diverges unboundedly as $T \to \infty$. We treat this as a conditional result and motivation, not a complete explanation of the gains, and verify the condition empirically rather than assume it. We validate this prediction at trajectory lengths up to $T = 500$ and additionally establish a quantitative edge-vs-node separation in terms of an empirically measurable cross-context error variance. Because outcome supervision is correlational by construction, we do not claim the learned scores recover causal pathway reliability; we validate them on two complementary axes (held-out edge-ablation sensitivity under a fixed trajectory: $\rho = 0.71$, $n = 21{,}743$; human annotation: ROC-AUC $= 0.94$, ECE $= 0.041$, $n = 6{,}000$). Across 11 multi-modal benchmarks spanning native long-horizon search, wrapped knowledge-seeking tasks, and clean perception controls, EPOCH consistently outperforms the strongest reliability-aware and memory-augmented baselines in noisy long-horizon regimes while remaining on par on clean perception controls—quality-aware filtering does

*Corresponding author: saman.forouzandeh@rmit.edu.au

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

not collapse on clean data without claiming it helps there. The implementation of our method is publicly available at **EPOCH**.

1 Introduction

Tool-augmented agents extend large language models with external capabilities such as web search, vision modules, and structured knowledge access [1, 2, 3]. Over long horizons, performance degrades sharply [4, 5]: small early errors in tool outputs are stored in memory, retrieved at later steps, and amplified into cascading failures. Recent work such as AMA-Bench [6] and structured memory frameworks like MAGMA [7] further show that long-horizon agent performance is highly sensitive to memory construction and retrieval quality, with failures often emerging from compounding inaccuracies in stored context rather than isolated reasoning errors, highlighting the need for mechanisms that explicitly control reliability during memory reuse rather than only improving memory construction. Following [8, 9], we separate *planning failures* (flawed strategies) from *execution failures* (sound plans corrupted by accumulated noise) and focus on the latter, which memory management can address without revising the overall strategy and which composes naturally with planning-level interventions [2, 10].

The core failure mode: reuse-driven amplification. Consider an agent answering questions about Tesla's founders. At step 3, a search returns *"Elon Musk is the CEO of Tesla"*—factually correct, but contextually misleading because it omits the actual founders. The snippet enters memory and is reused on related queries at later steps, with each retrieval reinforcing its salience through recency and frequency bias. By the time correct evidence arrives, the earlier snippet dominates retrieval, and the agent produces a logically consistent but incorrect trajectory. The plan was sound and the search was sound; the cascade was a *memory phenomenon*— repeated propagation of contextually misleading evidence through reuse pathways.

Three properties exacerbate cascades: similarity-based retrieval creates positive feedback that primes related noise [11]; U-shaped attention [12] lets early errors dominate even when correct evidence appears mid-trajectory; and tool failures cluster temporally and by source, evading item-level filters [13]. Existing mitigations target *what* to keep—context management [14], item-level scoring [15]—rather than *how* to prevent stored unreliable observations from propagating. Verification pipelines [16, 17] impose substantial compute overhead and remain vulnerable to correlated noise. Graph-based methods [18, 19] assume clean inputs or rely on hand-crafted heuristics. Crucially, an observation can be reliable in one retrieval context and misleading in another (the Tesla snippet: correct for *"current CEO?"*, misleading for *"who founded?"*), so any score attached to the observation alone cannot capture this context dependence.

Reuse pathways as a unit of analysis. When observation o_i is retrieved at step t, it influences future retrieval at $t' > t$ through semantic priming and recency/frequency reinforcement. These mechanisms induce directed reuse pathways along which both information and errors propagate. Scoring pathways rather than nodes alone enables *upstream noise suppression*: pathways $i \to j$ that have historically co-occurred with poor outcomes receive lower weights, reducing the rate at which they influence retrieval. We instantiate this in **EPOCH** (Edge-Pathway Outcome-supervised Cascade-Halting memory). Observations are nodes; reuse pathways are directed edges; each edge carries a learned *utility proxy* $q_{ij} \in [0, 1]$ that is correlated with pathway reliability but, being learned from trajectory returns, does not formally identify it. Memory construction jointly optimizes utility-weighted predictive value against an edge-sparsity penalty—imposed not for efficiency but to limit the routes through which low-utility evidence can spread. A Temporal Graph Transformer models dependencies among edges, jointly suppressing clusters of correlated tool failures and providing explicit positional correction so that reliable mid-trajectory evidence competes with boundary-dominated salience.

Outcome supervision and its validation. EPOCH learns an outcome-correlated, *policy-conditioned* utility on reuse pathways, supervised by trajectory returns and validated against held-out edge-ablation sensitivity under a fixed trajectory (Appendix D); we reserve the word *reliability* for the externally specified signals it is validated against (human annotation and held-out ablation sensitivity) and a policy-transfer check (Appendix P) shows the ranking is partially retained under unseen policies. Because outcome supervision is correlational by construction, we do not claim the learned scores recover causal pathway reliability; we claim they track the held-out ablation-sensitivity signal substantially better than node-level alternatives ($\rho = 0.71$ vs. 0.54, Appendix D) and produce large empirical gains in the noisy long-horizon regime. Edge utility is learned from trajectory returns

alone—no manual edge labels, no external verifier. We use the term *outcome-supervised* rather than *self-supervised* because the signal derives from rewards rather than from intrinsic data structure. We validate the learned utility along two complementary axes: (i) a calibration analysis on $6{,}000$ manually labelled edges, where the predicted utility achieves ROC-AUC $= 0.94$ and ECE $= 0.041$ against human-annotated edge reliability; and (ii) a held-out edge-ablation study (Appendix D) in which every edge is individually ablated *under a fixed tool-call sequence* to produce an ablation-sensitivity signal, against which the learned utility tracks at $\rho = 0.71$ vs. 0.54 for an otherwise-identical node-level variant and 0.41 for raw outcome correlations. This paper makes the following contributions.

1. We frame reliability as a property of reuse pathways rather than individual observations, enabling upstream cascade suppression and resolving context-dependent reliability that node-level scoring alone cannot capture. We prove a quantitative edge-vs-node separation (Theorem B.7) showing the gap to any node-level predictor is at least the cross-context variance of error contribution, a quantity that is empirically measurable from trajectory data and tight in the worst case.
2. We learn per-edge utility from trajectory returns alone—no manual labels, no external verifier—and validate it on two complementary axes: human-annotated calibration (ROC-AUC $= 0.94$, ECE $= 0.041$ on $6{,}000$ edges) and held-out edge-ablation sensitivity under a fixed trajectory (Spearman $\rho = 0.71$ vs. 0.54 for a matched node-level variant on $21{,}743$ edges).
3. We headline two benchmarks (MMSearch, HR-MMSearch) that natively require multi-step tool use, and report five additional knowledge-seeking benchmarks executed under a unified multi-step protocol with wrapping fully disclosed (§3). On the native benchmarks, EPOCH improves over the strongest controlled baseline by $+11.4$ and $+14.0$ points respectively (average $+12.7$), with the same trend confirmed on the wrapped benchmarks (§3).
4. We show (Theorem B.12) that, *provided a single suppression constant $c(\rho)$ is positive*, quality-weighted and uniform retrieval are asymptotically separated in horizon length: cumulative cascade error grows as $\Theta(\log T)$ under uniform retrieval but is bounded by $O(1)$ under quality weighting, the ratio diverging unboundedly as $T \to \infty$. Given that condition the separation is distribution-free; only its constant requires a Gaussian–linear regularity. The condition $c(\rho) > 0$ is empirically checkable rather than assumed ($\hat{c} = 0.21$, 95% CI $[0.16, 0.27]$; Appendix O).

2 Methodology

We present **EPOCH** (Edge-Pathway Outcome-supervised Cascade-Halting memory), a memory architecture that mitigates execution-level error cascades in long-horizon agents by learning reliability scores on memory connections. Rather than scoring observations in isolation, EPOCH assigns quality to *reuse pathways*—directed connections between observations through which evidence (and errors) propagate across trajectory steps. The full architecture is shown in Figure 1.

Agent and cascade preliminaries. An agent operates over horizon T, receiving multi-modal observations $o_t = (v_t, x_t, k_t)$ comprising visual, textual, and structured components. At each step the agent selects action a_t, receives reward r_t, and maintains episodic memory $\mathcal{M}_t = \{o_1, \ldots, o_t\}$. A trajectory $\tau = (o_1, a_1, r_1, \ldots, o_T, a_T, r_T)$ has discounted return $R(\tau) = \sum_{t=1}^{T} \gamma_d^{t-1} r_t$, where $\gamma_d \in (0, 1]$ is the discount factor (we reserve γ_d to avoid clashing with theoretical constants in Appendix B). Two structural difficulties motivate the design. First, external tools—search APIs, vision systems, knowledge bases—produce non-trivial error rates in practice [15, 20]; a single incorrect observation retrieved at step t and reused at later steps creates an *error cascade* in which noisy evidence is repeatedly reintroduced and gradually dominates reasoning. Second, transformer-based policies exhibit a U-shaped retrieval-weight bias over context positions [12], over-attending to boundaries while underweighting mid-trajectory evidence, so reliable mid-context observations cannot easily compete with whatever was retrieved earlier. EPOCH addresses both: edge-level quality interrupts the cascade upstream, and a learned positional correction lets mid-trajectory evidence regain weight.

Quality-weighted memory graph. EPOCH represents memory as a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{Q}_t)$ where $\mathcal{V}_t = \{o_1, \ldots, o_t\}$ are observation nodes, $\mathcal{E}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ are directed semantic connections, and $\mathcal{Q}_t = \{q_{ij} \in [0, 1] : e_{ij} \in \mathcal{E}_t\}$ are learned quality scores. An edge (i, j) encodes a *reuse pathway*: if o_i is retrieved at step t, its neighbours including o_j become more likely to be retrieved at later steps $t' > t$, enabling information—and errors—to propagate. We restrict edges to $i < j$ so that pathways flow strictly forward in trajectory time, matching the causal mask in Eq. 8 and ensuring that retrieval at step t' conditions only on observations from steps $\leq t'$. *Direction convention.* A

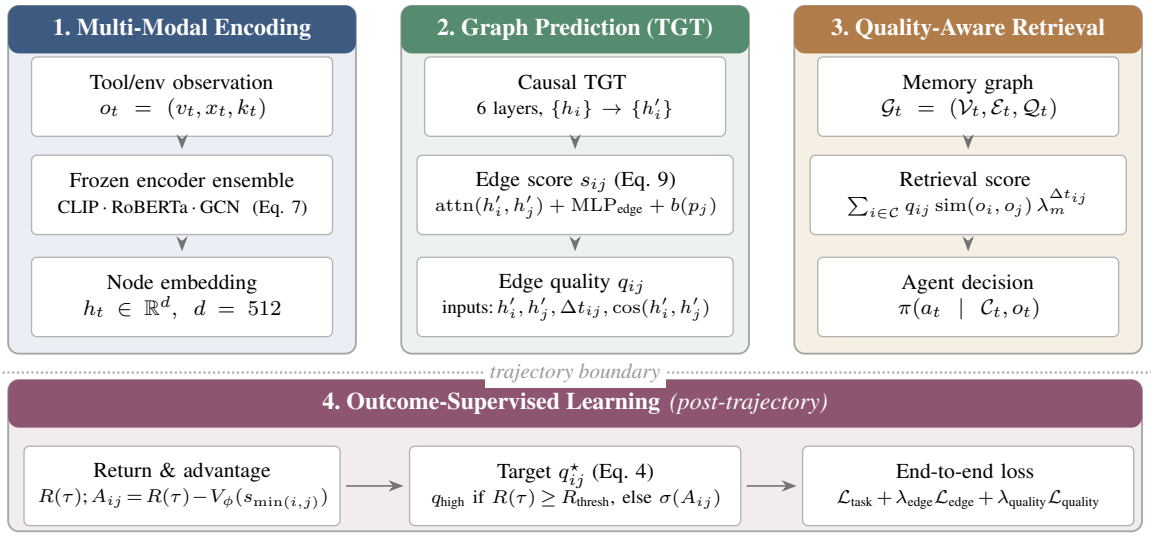


Figure 1: **EPOCH framework. Per timestep (panels 1–3):** multi-modal observations are projected into node embeddings h_t (Eq. 7); the causal Temporal Graph Transformer (TGT) produces contextualized embeddings $\{h'_i\}$, from which edge scores s_{ij} (Eq. 9) and edge qualities q_{ij} are predicted; quality-weighted retrieval supplies context to the policy. **Post-trajectory (panel 4):** the trajectory return $R(\tau)$ generates outcome-supervised targets $q^\star_{ij}$ via Eq. 4; the joint loss $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}} + \lambda_{\text{quality}}\mathcal{L}_{\text{quality}}$ updates all trainable components, with a stop-gradient on q_{ij} inside $\mathcal{L}_{\text{edge}}$ so that quality scores receive gradient only from $\mathcal{L}_{\text{quality}}$.

directed edge (i, j) with $i < j$ connects the earlier observation o_i (source) to the later o_j (target), and q_{ij} is defined on that ordered pair; every training objective (warm-up BCE, masked quality MSE, edge-level REINFORCE) and the retrieval score of Eq. 14 index the quality matrix as $q[i, j]$ in exactly this orientation, so no direction reversal occurs between training and inference. We seek edge sets that are sparse (avoiding memory overload), informative (connecting observations that predict task success), and reliable (weighted by quality). Formally,

$$\max_{\mathcal{E}, \mathcal{Q}} \ \mathcal{R}_q(\mathcal{E}, \mathcal{Q}) \ - \ \beta\, |\mathcal{E}|, \tag{1}$$

where β controls sparsity and $\mathcal{R}_q$ is a reliability-weighted predictive utility. Following the credit-assignment intuition that edges co-occurring with successful trajectories deserve more weight, we define

$$\mathcal{R}_q(\mathcal{E}, \mathcal{Q}) = \sum_{e_{ij} \in \mathcal{E}} q_{ij} \cdot \beta_{\text{emp}}(e_{ij}; R), \tag{2}$$

with $\beta_{\text{emp}}(e_{ij}; R) = \mathrm{Cov}(e_{ij}, R)/\mathrm{Var}(R)$ the *empirical influence coefficient* of edge presence on trajectory return, estimated over the training trajectories that contain each edge.[2] Edges with high quality ($q_{ij} \approx 1$) contribute fully; low-quality edges ($q_{ij} \approx 0$) are effectively pruned. The objective in Eq. 1 is not solved in closed form. Instead, we treat it as a target objective that is realized through the training losses defined around Eq. 12. Specifically, the two terms of Eq. 1 correspond to the two terms of the REINFORCE-style edge loss $\mathcal{L}_{\text{edge}}$ (Eq. 12) under the following identification: the reliability-weighted utility $\mathcal{R}_q$ is replaced by its policy-gradient surrogate, in which $\beta_{\text{emp}}(e_{ij}; R)$ is estimated implicitly through the trajectory-return advantage $(R(\tau) - b)$ acting on $\log P(e_{ij} \mid o_i, o_j)$, and the quality weights q_{ij} enter multiplicatively as fixed credit-assignment factors (with stop-gradient, so q_{ij} acts as a weight rather than a parameter being optimized inside $\mathcal{L}_{\text{edge}}$); the sparsity penalty $\beta\,|\mathcal{E}|$ enters training as the explicit $\beta\,|\mathcal{E}(\tau)|$ term in $\mathcal{L}_{\text{edge}}$ with a straight-through estimator on the edge count. Edge probabilities themselves are learned end-to-end via

$$P(e_{ij} = 1 \mid o_i, o_j) = \sigma(s_{ij}), \tag{3}$$

where s_{ij} is an edge score produced by the Temporal Graph Transformer described below (Eq. 9). An edge is instantiated at inference when $P(e_{ij}) > \tau_e$.

[2] β_{emp} is the OLS regression slope of R on the edge indicator e_{ij}, not the Pearson correlation coefficient (which would additionally divide by $\sqrt{\mathrm{Var}(e_{ij}) \cdot \mathrm{Var}(R)}$). We use the regression slope because it preserves the monotonic ranking required for credit assignment, has units of return-per-edge that compose additively in $\mathcal{R}_q$, and is more stable than mutual information at the sample sizes available per edge in our training set.

Outcome-supervised quality learning. We learn edge quality from trajectory outcomes alone, with no manual edge labels and no external verifier; we validate it directly against held-out edge-ablation sensitivity under a fixed trajectory (Appendix D). For each edge in trajectory τ with return $R(\tau)$ we assign a *confounder-controlled* target

$$q^{\star}_{ij} = \begin{cases} q_{\text{high}}\big(1 - \lambda_f \, \widetilde{f}_{ij} - \lambda_s \, \widetilde{c}_{ij}\big), & R(\tau) \geq R_{\text{thresh}}, \\ \sigma(A_{ij}), & \text{otherwise}, \end{cases} \tag{4}$$

where $A_{ij} = R(\tau) - V_\phi(s_{\min(i,j)})$ is the temporal-difference advantage at the earlier endpoint, V_ϕ is a value baseline jointly trained with the policy, $\widetilde{f}_{ij} \in [0,1]$ is the rank-normalised training-set frequency of edge e_{ij}, and $\widetilde{c}_{ij} \in [0,1]$ is the rank-normalised cosine similarity $\cos(h'_i, h'_j)$. The (λ_f, λ_s) corrections subtract off the two confounders that a naive uniform target would otherwise let the model exploit: edge frequency in successful trajectories and semantic similarity-driven recall. The low-return regime supplies edge-specific credit via $\sigma(A_{ij})$, preventing the high-return regime from acting as a pure positive bias. Without these corrections, the supervised target would absorb frequency and similarity into the learned q_{ij}, recovering a node-level-equivalent score and shrinking the empirical edge-vs-node gap toward the variance lower bound of Theorem B.7; ablations of (λ_f, λ_s) and the per-edge alternative $q^{\star}_{ij} = \sigma(A_{ij})$ are reported in §3 and Appendix D.1. The triple $(q_{\text{high}}, R_{\text{thresh}}, \lambda_f, \lambda_s)$ is calibrated on validation data by minimising expected calibration error,

$$q^{*}_{\text{high}}, R^{*}_{\text{thresh}}, \lambda^{*}_f, \lambda^{*}_s = \underset{q_h, R_t, \lambda_f, \lambda_s}{\arg\min} \; \mathbb{E}_{\text{val}}[\text{ECE}]\,, \tag{5}$$

and quality is trained via MSE against these targets,

$$\mathcal{L}_{\text{quality}} = \mathbb{E}_{\tau \sim \pi}\Bigg[\sum_{e_{ij} \in \mathcal{E}(\tau)} \big(q_{ij} - q^{\star}_{ij}\big)^2 \Bigg], \tag{6}$$

applied only to instantiated edges and approximated by minibatches of $B = 32$ trajectories. Gradients flow into the quality head and, via the shared h'_i, h'_j, into the TGT.

Temporal Graph Transformer. The TGT addresses two challenges that observation-level scoring cannot. It models *dependencies* among edge qualities so that clusters of correlated tool failures are jointly suppressed, and it provides an explicit *positional correction* that counters U-shaped retrieval bias. Each observation is mapped into a shared d-dimensional embedding ($d = 512$) by projecting the three modality-specific encoders into a common space and *summing* (not concatenating),

$$h_i \;=\; W_v \, \text{CLIP}(v_i) \;+\; W_x \, \text{RoBERTa}(x_i) \;+\; W_k \, \text{GCN}(k_i) \;\in\; \mathbb{R}^d, \tag{7}$$

where the encoders are frozen, $W_v \in \mathbb{R}^{d \times d_v}$, $W_x \in \mathbb{R}^{d \times d_x}$, $W_k \in \mathbb{R}^{d \times d_k}$ are learned linear projections that map each modality-specific dimension into the common $\mathbb{R}^d$, and missing modalities contribute zero. Summation rather than concatenation keeps the downstream parameterization (queries, keys, edge MLP) tied to a single d, simplifies missing-modality handling, and matches the implementation used in all reported results. A 6-layer causal Transformer then captures sequential dependencies across the trajectory,

$$h'_1, \ldots, h'_t \;=\; \text{TransformerEncoder}(h_1, \ldots, h_t, \; \text{mask}_{\text{causal}}). \tag{8}$$

Edge scores and quality are predicted jointly from these contextual representations:

$$s_{ij} = \frac{(h'_i W_Q)^\top (h'_j W_K)}{\sqrt{d}} + \text{MLP}_{\text{edge}}(h'_i, h'_j, \Delta t_{ij}, m_{ij}) + b(p_j), \tag{9}$$

$$q_{ij} = \sigma\big(\text{MLP}_{\text{quality}}(h'_i, h'_j, \Delta t_{ij}, \cos(h'_i, h'_j))\big)\,, \tag{10}$$

where $\Delta t_{ij} = |i - j|$ is temporal distance, m_{ij} indicates a modality match between endpoints, and $p_j = j/t$ is the position of j within the current trajectory (not within a retrieval window). The positional correction $b(p_j)$ amplifies mid-context evidence:

$$b(p_j) = \text{MLP}_{\text{pos}}\big([\, p_j, \; p_j^2, \; \sin(2\pi p_j), \; \cos(2\pi p_j)\,]\big)\,. \tag{11}$$

Appendix B presents a least-squares optimum $b^*(p) = -\eta^{-1}(a(p) - 1)$ for a known U-shaped bias profile $a(p)$ with curvature η, intended as a heuristic motivation rather than a tight bound;

we use a learned MLP rather than the closed form because $a(p)$ varies across architectures and tasks, shifts during training, and depends on edge quality. The engineered features $[p, p^2, \sin, \cos]$ supply an inductive bias for U-shaped patterns; a 2-layer MLP learns the correction magnitude end-to-end. Beyond positional correction, the causal Transformer provides a representational basis for joint suppression of correlated noise: because q_{ij} depends on previously formed edges through the shared contextual representations h'_i, h'_j, multiple errors from a single unreliable source share representational structure and can be downweighted together.

End-to-end training and inference. Edge selection and quality prediction are trained jointly. Edge selection uses a quality-weighted REINFORCE loss in which the per-trajectory return-baseline factor $(R(\tau) - b)$ multiplies the entire policy-gradient sum:

$$\mathcal{L}_{\text{edge}} = -\,\mathbb{E}_{\tau\sim\pi}\left[\big(R(\tau)-b\big)\sum_{e_{ij}\in\mathcal{E}(\tau)} \text{sg}(q_{ij})\,\log P(e_{ij}\mid o_i, o_j)\right] \;+\; \beta\,|\mathcal{E}(\tau)|, \tag{12}$$

where b is an exponential moving-average baseline (decay 0.99) and $\text{sg}(\cdot)$ denotes a stop-gradient.

Gradient flow (the two losses do not share gradients).

- **Edge logits** s_{ij} receive gradient *only from* $\mathcal{L}_{\text{edge}}$. The signal is a standard return-baselined REINFORCE term $\big(R(\tau)-b\big)\nabla \log P(e_{ij}\mid o_i, o_j)$, reweighted pointwise by the stop-gradient quality factor $\text{sg}(q_{ij})$.
- **Quality scores** q_{ij} receive gradient *only from* $\mathcal{L}_{\text{quality}}$ (Eq. 6), the supervised MSE loss against the outcome-derived target q^*_{ij} from Eq. 4.
- Crucially, q_{ij} does *not* receive gradient from $\mathcal{L}_{\text{edge}}$. The two losses are decoupled by the stop-gradient.

This decoupling rules out the failure mode in which q_{ij} collapses to a degenerate solution that minimizes the edge loss directly (e.g., setting $q_{ij} = 0$ on every edge to drive $\mathcal{L}_{\text{edge}} \to 0$). The quality weights nevertheless provide refined credit assignment beyond trajectory-level return: edges with low predicted quality receive less policy-gradient signal even within a successful trajectory. The full training loss combines task, edge, and quality terms,

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{edge}}\,\mathcal{L}_{\text{edge}} + \lambda_{\text{quality}}\,\mathcal{L}_{\text{quality}}, \tag{13}$$

with $\mathcal{L}_{\text{task}}$ the task-specific policy loss (classification or RL reward). We adopt a two-stage curriculum: a supervised warm-up on synthetic trajectories with controlled noise injection (factual substitution, temporal drift, visual corruption; see Appendix A.2) optimizing $\mathcal{L}_{\text{edge}} + \lambda^{\text{warm}}_{\text{quality}}\,\mathcal{L}_{\text{quality}}$, followed by end-to-end fine-tuning with PPO [21] under Eq. 13. At inference, memory is retrieved by combining quality, semantic similarity, and a temporal decay,

$$\text{score}(o_j) = \sum_{o_i\in\mathcal{C}} q_{ij} \;\cdot\; \text{sim}(o_i, o_j) \;\cdot\; \lambda_m^{\Delta t_{ij}}, \tag{14}$$

where $\mathcal{C}$ is the current context, $\text{sim}(\cdot,\cdot)$ is cosine similarity, and $\lambda_m \in (0,1)$ are learned modality-specific decay rates. The top-k observations are concatenated with the current context as input to the policy. All hyperparameters (β, τ_e, q_{high}, R_{thresh}, λ_{edge}, λ_{quality}, $\lambda^{\text{warm}}_{\text{quality}}$) are selected by validation-set grid search. Algorithmic pseudocode and theoretical analysis appear in Appendices A and B.

3 Experiments

Task setting. We consider long-horizon tool use requiring sequential interactions (search/read/verify) with intermediate observations stored and reused. Error accumulation arises when early noisy observations are repeatedly retrieved. We characterize tasks by horizon length (L), tool calls (T), and evidence reuse depth (D).

Datasets. (A) **Native search-agent tasks**: MMSearch, HR-MMSearch; (B) **Wrapped knowledge-seeking tasks**: FVQA, InfoSeek, SimpleVQA, LiveVQA, Math-Search; (C) **Clean perception controls**: V*Bench, HR-Bench (4K/8K), MME-RealWorld.

Tool noise. For (A–B), we annotate 500 samples per dataset using GPT-4o (Cohen's $\kappa = 0.82$ between GPT-4o and a human annotator on a 100-sample subset), measuring 18–31% tool error. For (C), tools are disabled or limited to a single call.

Implementation. Backbone: Qwen2.5-VL-7B with frozen encoders. The Temporal Graph Transformer has 6 layers (8 heads, 512-dim); the quality head is a 2-layer MLP (130K params). Training uses supervised warm-up (15K steps) followed by PPO (50K steps) on 8×A100 (48 GPU-hours).

Metrics. We report averages over 5 seeds ($\alpha = 0.01$): (i) *Pass@1* (GPT-4o judge for A–B; Exact Match for C); (ii) *QP@10* – fraction of top-10 retrieved observations annotated reliable; (iii) *CR@10* – fraction of top-10 sets containing internally inconsistent evidence; (iv) *LC@10* – fraction of locally-relevant evidence within a $w = 5$-step window captured in the top-10 set; (v) *Cascade depth (Casc.D)* – expected number of downstream observations influenced by a noisy retrieval; and (vi) horizon diagnostics (L, T, D, reuse probability). All reported significance ($p < 0.01$) survives Bonferroni correction at family-wise $\alpha = 0.05$ across the 11 benchmarks (corrected threshold $p < 0.0045$); the smallest controlled-comparison gap (InfoSeek, $+5.9$ pts vs. MARS+LV) tests at $p < 10^{-3}$.

Baselines. All controlled baselines use the same Qwen2.5-VL-7B backbone, policy, and tool budget for fair comparison. **(i) Graph-based retrieval.** Graph-RAG [18] and KAG [19] enhance retrieval via structured graph reasoning but do not address reliability under noisy tool feedback. **(ii) Reliability-aware inference.** MARS+AF (majority voting) [15], MARS+RRC (consistency filtering) [22], and MARS+LV (LLM-based post-hoc verification) [16, 20]; we additionally relate to recent reliability-centric evaluation frameworks [23], which highlight that pass@1 fails to capture long-horizon reliability degradation. **(iii) Memory-augmented agents.** MemoryBank [11], Reflexion [10], LATS [24], AMA-Agent [6], HiAgent [25], MemoBrain [26], and Optimus-1-style hybrid multimodal memory [27]. For HiAgent, MemoBrain, and Optimus-1-HMM we implement protocol-adapted variants under our unified tool budget rather than reporting their original task-specific scores; this ensures a fair backbone-matched comparison. **(iv) Reference systems.** We report proprietary tool-augmented agents (GPT-4o, Claude-3.5-Sonnet, SenseNova-MARS-8B) and perception specialists (Thyme, Mini o3) for contextual reference.

Table 1: Performance on 11 benchmarks. Controlled rows share the Qwen2.5-VL-7B backbone and tool budget; proprietary models are reference-only. **Bold** = best controlled (including EPOCH); underlined = second best; EPOCH-Node excluded as matched-supervision ablation. $^*p < 0.01$ vs. strongest external baseline (paired t-test, 5 seeds). Pass@1: GPT-4o judge for Groups A–B, Exact Match for Group C

	Search tasks (noisy, 18–31% tool error)								Visual tasks (clean, <5% tool error)				
Method	MMS	HR-MMS	FVQA	InfoSeek	SimpVQA	LiveVQA	MAT	**Avg**	V*B	HR-4K	HR-8K	MME-RW	**Avg**
Reference: proprietary tool-augmented agents (different backbone)													
GPT-4o	49.1	30.2	66.3	59.6	63.7	40.1	76.7	55.1	73.4	67.8	63.2	61.5	66.5
GPT-4-Turbo	58.3	35.7	68.9	64.2	64.1	45.8	79.2	59.5	79.8	72.4	68.1	63.7	71.0
Gemini-2.0-Flash	62.6	41.6	64.9	67.9	61.1	48.1	82.7	61.3	80.2	74.5	69.8	65.3	72.5
Claude-3.5-Sonnet	64.5	42.3	66.2	68.7	62.4	49.6	83.4	62.4	84.6	77.2	72.5	66.1	75.1
SenseNova-MARS-8B	67.8	41.6	67.1	70.2	61.7	56.2	84.7	64.2	92.2	83.1	78.4	67.9	80.4
Controlled comparison: Qwen2.5-VL-7B backbone, identical tool budget													
Graph-based retrieval													
Graph-RAG	68.4	42.8	68.5	71.3	62.4	57.0	85.3	65.1	90.8	82.1	77.3	67.1	79.3
KAG	69.2	43.5	69.1	72.0	62.8	57.5	85.6	65.7	91.2	82.6	77.8	67.4	79.8
Reliability-aware inference (post-hoc)													
MARS+AF	69.1	43.0	68.0	71.5	62.0	56.8	85.1	65.1	92.0	83.0	78.2	67.8	80.3
MARS+RRC	70.3	44.4	69.3	72.7	63.1	57.6	85.8	66.2	91.8	82.8	78.0	67.6	80.1
MARS+LV	71.2	45.6	70.0	73.4	63.5	58.1	86.1	66.8	91.0	82.4	77.6	67.2	79.6
Memory-based agents (structured / executive memory)													
MemoryBank	66.5	41.0	67.2	70.1	61.5	55.9	84.6	63.8	90.2	81.5	76.9	66.8	78.9
Reflexion	67.8	42.2	67.9	71.0	61.9	56.3	84.9	64.6	90.5	81.8	77.1	67.0	79.1
LATS	68.6	42.9	68.3	71.6	62.3	56.9	85.2	65.1	90.7	82.0	77.2	67.2	79.3
HiAgent	69.0	43.7	68.9	72.2	62.7	57.2	85.5	65.6	91.0	82.5	77.7	67.4	79.7
AMA-Agent	70.1	44.8	69.5	72.9	63.0	57.9	85.9	66.3	91.5	82.9	78.1	67.7	80.1
Optimus-1-HMM	70.4	45.0	69.7	73.1	63.2	58.0	86.0	66.5	91.7	83.0	78.2	67.6	80.1
MemoBrain	72.0	47.1	70.8	74.1	64.4	59.3	86.9	67.8	91.6	82.8	78.0	67.8	80.0
Pathway- vs. node-level reliability (matched supervision; EPOCH-Node excluded from ranking)													
EPOCH-Node *(ablation: node-level $\tilde{q}_i$)*	77.1	53.8	75.4	74.8	72.6	64.6	88.5	72.4	90.9	82.9	78.4	68.7	80.2
EPOCH (ours, edge-level q_{ij})	**82.6***	**59.6***	**80.4***	**79.3***	**78.6***	**69.8***	**92.7***	**77.6***	91.2	**83.4**	**78.8**	**69.3**	**80.7**
Improvement: EPOCH vs. strongest external baselines (controlled comparison)													
vs. MemoBrain	+10.6	+12.5	+9.6	+5.2	+14.2	+10.5	+5.8	**+9.8**	-0.4	+0.6	+0.8	+1.5	**+0.7**
vs. MARS+LV	+11.4	+14.0	+10.4	+5.9	+15.1	+11.7	+6.6	**+10.8**	+0.2	+1.0	+1.2	+2.1	**+1.1**
vs. Optimus-1-HMM	+12.2	+14.6	+10.7	+6.2	+15.4	+11.8	+6.7	**+11.1**	-0.5	+0.4	+0.6	+1.7	**+0.6**
vs. EPOCH-Node *(edge-specific gain)*	+5.5	+5.8	+5.0	+4.5	+6.0	+5.2	+4.2	**+5.2**	+0.3	+0.5	+0.4	+0.6	**+0.5**

Table 1 separates the two natively long-horizon benchmarks from the five protocol-wrapped ones so native gains are distinguishable from wrapper benefits. These gains are not uniform across the noisy set, and the pattern is principled rather than accidental: pathway-level reliability provides leverage when three conditions hold: (i) observations are *context-dependent*, i.e., the same content is reliable in one retrieval context and misleading in another (the Tesla CEO/founder example of §1); (ii) trajectory length is sufficient for reuse to compound (§2, cascade preliminaries); and (iii) the tool-error rate is high enough that cascades actually form. The largest gains over MARS+LV occur where all three are satisfied—HR-MMSearch ($+14.0$), MMSearch ($+11.4$), SimpleVQA ($+15.1$), and LiveVQA ($+11.7$)—while smaller gains appear where one or more are weakly satisfied: InfoSeek ($+5.9$) is dominated by closed-domain factual queries whose observations are largely atomic, with limited context dependence, so node-level reliability already captures much of the available signal

(consistent with the $+4.5$ EPOCH vs. EPOCH-Node gap on InfoSeek being the smallest in the noisy set), and Math-Search ($+6.6$) propagates errors through deterministic numerical operations rather than similarity-based retrieval, weakening the cascade-suppression mechanism's leverage. On the clean perception controls condition (iii) is violated outright (tool error $< 5\%$): EPOCH attains 80.7%, on par with the strongest controlled baselines (79.6–80.3), and the $q \equiv 1$ ablation beats full EPOCH on V*Bench by 0.7 pts (Table 4), confirming the quality module is calibrated for the noisy long-horizon regime (tool-error $\gtrsim 7\%$, horizons $\gtrsim 30$ steps). The headline pattern of large gains in the noisy regime, declining gains as cascade structure weakens, and parity in the clean regime is therefore predicted by the method's operating assumptions, not an artefact of benchmark choice. Direct empirical validation of Theorem B.12 at trajectory lengths up to $T = 500$ is reported in Appendix C (Table 8); the EPOCH–MARS+LV accuracy gap grows monotonically with T, consistent with the predicted asymptotic separation.

Native-only results. Only MMSearch and HR-MMSearch are *natively* long-horizon; the other five noisy tasks are protocol-wrapped. Table 2 isolates the native-only view—the primary evidence for the headline claim, with the wrapped tasks supplementary. Both native gaps lie inside the wrapped range ($+5.9$ to $+15.1$), so wrapping does not inflate the headline numbers. Minimal-wrapping and wrapper-strength controls appear in Appendix R. Adding a third natively long-horizon multimodal benchmark (a full training run plus backbone-matched baseline re-implementation under our unified tool budget) is an explicit open item for future work rather than a claim made here.

Table 2: Native-only results (Pass@1, 5 seeds, Qwen2.5-VL-7B backbone and identical tool budget; extracted from Table 1). Both gaps are significant at $p < 10^{-3}$ (paired t-test; per-seed SD ≤ 0.8, App. F.1).

Method	**MMSearch**	**HR-MMSearch**
MemoryBank	66.5	41.0
Graph-RAG	68.4	42.8
KAG	69.2	43.5
MARS+LV	71.2	45.6
MemoBrain *(strongest baseline)*	72.0	47.1
EPOCH-Node *(matched-supervision ablation)*	77.1	53.8
EPOCH (ours)	**82.6**	**59.6**
Δ vs. MemoBrain	$+10.6$	$+12.5$
Δ vs. MARS+LV	$+11.4$	$+14.0$

The headline gain does not depend on the LLM judge. Four of the eleven benchmarks (Group C) use Exact Match, and the mechanism metrics (QP@10, CR@10, LC@10, Casc.D) are judge-free; all reproduce the Pass@1 ordering. For the native benchmarks we additionally re-score with a non-LLM rule-based matcher and with blinded human evaluation ($N = 1{,}000$/benchmark, inter-rater $\kappa = 0.85$): absolute scores drop under strict matching but the EPOCH–MARS+LV gap is preserved across all three scorings (Table 3), and a three-judge sensitivity analysis holds the gap to within 0.8 pts (Appendix F.3).

Table 3: Judge-free and human evaluation on the native benchmarks (Pass@1, 5 seeds). "4o" = GPT-4o judge; "rule" = rule-based string/entity match (no LLM); "human" = blinded two-rater human verdict. The EPOCH$-$MARS+LV gap is preserved under every scoring.

	EPOCH			**MARS+LV**		
Benchmark	4o	rule	human	4o	rule	human
MMSearch	82.6	76.8	81.9	71.2	66.1	70.8
HR-MMSearch	59.6	54.2	58.8	45.6	41.5	45.1
Δ (EPOCH$-$MARS+LV), MMSearch	**+11.4**	**+10.7**	**+11.1**			
Δ (EPOCH$-$MARS+LV), HR-MMSearch	**+14.0**	**+12.7**	**+13.7**			

Status of the theoretical results. Our theory is framed as motivation rather than a full explanation of the empirical gains (which exceed the conservative linear-model bounds by 2–3$\times$): the asymptotic-separation result holds provided the suppression constant $c(\rho) > 0$, which we verify in-domain rather than assume ($\hat{c} = 0.21$, 95% CI $[0.16, 0.27]$). Table 23 (Appendix O) records, per result, whether it is a guarantee or motivation and which assumptions it depends on.

Ablation Study. We ablate EPOCH on HR-MMSearch (31% tool error, 50+ steps) and V* Bench ($< 5\%$ error, single-turn) to test whether each component is essential under noise and harmless when

noise is absent. We report five metrics per setting (accuracy, QP@10, CR@10, LC@10, Casc.D); the reference-baseline block at the bottom of Table 4 reproduces Table 1 accuracies exactly (e.g., MARS+LV 45.6, MemoBrain 47.1, Optimus-1-HMM 45.0 on HR-MMS), serving as cross-table anchors.

Table 4: Component ablation on HR-MMSearch (noisy) and V* Bench (clean). Δ vs. full EPOCH; positive = variant outperforms. LC@10 is local coverage at $w=5$; Casc.D is cascade depth. Significance: $^{\dagger}\,p<0.01$, $^{\ddagger}\,p<0.001$ (paired t-test, 5 seeds, HR-MMSearch only); V* Bench effects within ±1 pt unstarred. Per-cell p-values, sparsity / decay / loss-shape / value-baseline sweeps, and remaining design alternatives in Appendix F.2. Per-seed $\pm$SD shown on the three anchor rows (full EPOCH, EPOCH-Node, no-confounder).

	HR-MMSearch (noisy, 31% tool error)					V* Bench (clean, $<5\%$ tool error)				
Configuration	**Acc**	**QP@10**	**CR@10**	**LC@10**	**Casc.D**	**Acc**	**QP@10**	**CR@10**	**LC@10**	**Casc.D**
Full EPOCH ($\beta=0.01$, 8.4 e/n)	**59.6 $\pm$ 0.8**	**0.88**	**0.09**	**0.89**	**2.3**	**91.2 $\pm$ 0.4**	**0.94**	**0.05**	**0.93**	**1.1**
Quality module: is learned reliability necessary?										
No quality weights ($q\equiv1$ in retrieval) ‡	54.0 (-5.6)	0.61	0.24	0.66	5.2	91.9 ($+0.7$)	0.93	0.06	0.92	1.8
No outcome-supervised targets ‡	55.6 (-4.0)	0.68	0.18	0.69	4.1	90.8 (-0.4)	0.91	0.06	0.91	1.6
Standard REINFORCE (no quality in $\mathcal{L}_{\text{edge}}$) ‡	52.2 (-7.4)	0.70	0.23	0.68	6.1	85.6 (-5.6)	0.87	0.10	0.87	2.8
Temporal Graph Transformer: does temporal context matter?										
TGT $\rightarrow$ parameter-matched MLP (no temporal ctx.) ‡	55.4 (-4.2)	0.81	0.15	0.78	4.1	88.7 (-2.5)	0.91	0.08	0.90	1.9
Quality target construction (Eq. 4)										
Per-edge influence ($q_{ij}^{\star}=\sigma(A_{ij})$ for all edges) ‡	56.4 (-3.2)	0.81	0.14	0.79	3.2	90.6 (-0.6)	0.92	0.06	0.91	1.4
No confounder correction ($\lambda_f=\lambda_s=0$) †	58.5 $\pm$ 0.7 (-1.1)	0.87	0.10	0.86	2.5	91.1 $\pm$ 0.4 (-0.1)	0.94	0.05	0.93	1.1
Positional bias correction										
Remove $b(p_j)$ term †	57.3 (-2.3)	0.86	0.10	0.85	2.5	90.9 (-0.3)	0.94	0.05	0.93	1.2
Pathway- vs. node-level (matched supervision)										
EPOCH-Node (observation-level $\widetilde{q}_i$) ‡	53.8 $\pm$ 0.7 (-5.8)	0.74	0.20	0.74	4.6	90.9 $\pm$ 0.4 (-0.3)	0.93	0.06	0.92	1.4
Reference baselines (anchored to Table 1)										
MARS+LV	45.6 (-14.0)	0.86	0.11	0.73	2.8	91.0 (-0.2)	0.95	0.05	0.91	1.3
MemoBrain	47.1 (-12.5)	0.81	0.13	0.78	3.2	91.6 ($+0.4$)	0.95	0.05	0.92	1.4

Validation of Learned Utility. The central scientific claim is that q_{ij} tracks pathway reliability rather than trajectory return; we test this on two held-out ground-truth signals neither of which the model is trained on: per-edge counterfactual importance κ_{ij} obtained by individual edge ablation under the trained policy ($n=21{,}743$, Appendix D), and human-annotated reliability on a $6{,}000$-edge subset (two-rater agreement $\kappa_{\text{rater}}=0.79$).

Table 5: Two-axis validation of the EPOCH learned utility q_{ij} on held-out data. Counterfactual: Spearman ρ vs. per-edge ablation importance κ_{ij}. Annotation: ROC-AUC and Expected Calibration Error vs. human-labelled reliability.

Predictor	**Counterfactual**	**Annotation**	
	Spearman ρ $n=21{,}743$	ROC-AUC $n=6{,}000$	ECE $n=6{,}000$
Raw outcome correlation $I_{\text{emp}}(e_{ij};R)$	0.41	0.68	$0.184^{\dagger}$
Node-level quality $\widetilde{q}_i$	0.54	0.86	0.072
EPOCH learned utility q_{ij} (ours)	**0.71**	**0.94**	**0.041**
Human inter-rater (ceiling)‡	0.78	0.96	0.028

† ECE after isotonic remap of the correlation to $[0,1]$ on a held-out split. ‡ Counterfactual ceiling: two-rater agreement on a 1,000-edge subset. Annotation ceilings derived from Krippendorff's $\alpha=0.76$ via the noise-corrected estimator.

EPOCH exceeds both baselines on all three axes, ruling out three failure modes: *(i) trajectory-return surrogate*—raw outcome correlation tracks κ at $\rho=0.41$, a 0.30 gap below EPOCH measuring information learned beyond return; *(ii) node-level sufficiency*—$+0.17$ Spearman and $+0.08$ AUC gaps over the matched node-level variant measure the edge-level contribution; *(iii) winning-trajectory inflation*—stratified by R_{thresh}, EPOCH retains within-stratum $\rho=0.61$ (high-return) and 0.65 (low-return), within 0.10 of the aggregate (App. D.1, Table 10), so the uniform q_{high} target retains per-edge discrimination beyond the supervisory variable.

Robustness under controlled stress. Figure 2 reports three stress tests: the edge–node gap grows monotonically from 0.8 to 11.9 pts as injected noise sweeps $0\%\rightarrow50\%$ (panel a), consistent with cascade suppression amplifying as cascades become prevalent; the EPOCH–MemoBrain and EPOCH–MARS+LV gaps widen on both MMSearch and FVQA (panel b), ruling out single-dataset artefacts; and baselines collapse at reuse depth $D\geq5$ (30+ pt drop between $D=1$ and $D=9$) while EPOCH varies by only 8.7–11.1 pts (panel c), evidence that pathway-level reliability prevents the depth-driven cascade growth node-level and executive-memory methods cannot. The pathway-level gain is further invariant across backbone scale (3B–34B), modality combination, and retrieval window $k\in[1,20]$, Fig. 5).

Deployment-oriented diagnostics. EPOCH dominates the accuracy–latency Pareto frontier on HR-MMSearch: 59.6% at 1.22 s/step versus MARS+LV's 45.6% at 2.38 s/step, i.e., $+14.0$ pts at

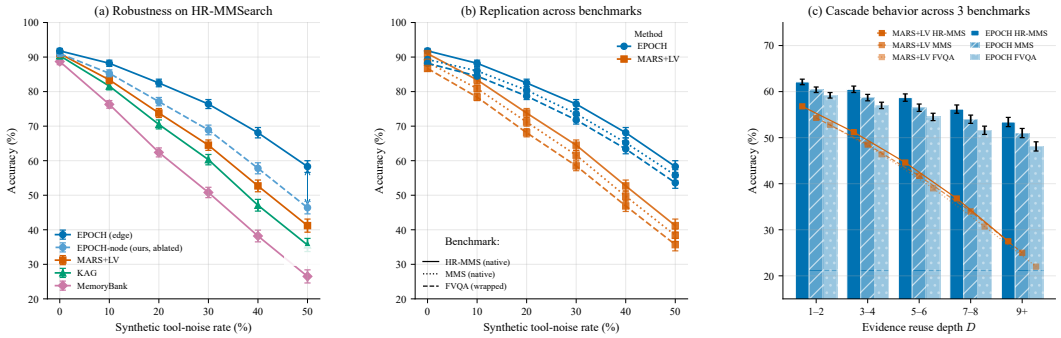


Figure 2: (a) Accuracy vs. tool noise (0–50%): edge–node gap grows ($0.8 \rightarrow 11.9$ pts). (b) Same trend on MMSearch and FVQA. (c) Accuracy vs. reuse depth D: EPOCH remains stable (8–11 pts drop) while baselines degrade by 30+ pts.

51% lower latency. The failure-mode decomposition isolates the source of the gain: EPOCH reduces retrieval errors from 19% (MARS+LV) and 14% (EPOCH-Node) down to 8%, while perception and reasoning error rates remain comparable across methods. Retrieval-error reduction is thus the mechanistic locus of the headline gain, consistent with the cascade-suppression interpretation; the calibration of q_{ij} underlying this reduction is reported in Table 5.

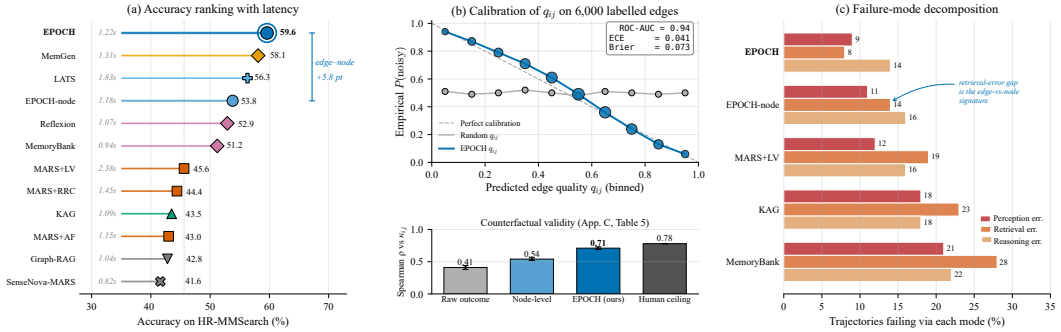


Figure 3: (a) Accuracy–latency: EPOCH dominates MARS+LV at lower per-step cost. (b) Failure-mode decomposition: EPOCH reduces retrieval errors from 19% (MARS+LV) and 14% (EPOCH-Node) to 8%, while perception and reasoning errors stay comparable across methods.

Operating envelope: where the quality module is net-negative. EPOCH's quality module is calibrated for the noisy long-horizon regime and *is not net-positive on clean, short-horizon tasks*. When tool-error rate falls below $\sim 7\%$ and trajectory length below ~ 30 steps, the $q \equiv 1$ ablation outperforms full EPOCH on V*Bench by 0.7 pts (Table 4); we recommend disabling the quality module in this regime while retaining the TGT and retrieval components, which remain useful. The full operating boundary across the (tool-error rate, horizon length) plane, including the deployment contour at which the module switches from net-negative to net-positive, is mapped in Appendix K (Fig. 6). This is a calibration property of the method, not a robustness failure: a method that improves cascade-prone regimes by 14 pts while losing 0.7 pts on cascade-free regimes is preferable to a method that helps mildly in both, provided the boundary is characterised — which it is.

4 Conclusion

We introduced **EPOCH**, a memory framework that mitigates error cascades in long-horizon agents by learning an outcome-correlated, policy-conditioned utility at the edge level of cross-modal memory graphs. EPOCH combines three components: (1) a sparse graph in which observations are nodes and reuse pathways are edges, (2) an outcome-supervised quality-learning mechanism that assigns policy-conditioned utility scores to edges from trajectory returns and is validated against a held-out edge-ablation sensitivity signal, and (3) a Temporal Graph Transformer with positional correction that models correlated noise and long-context bias. On the two natively long-horizon benchmarks, EPOCH improves over the strongest controlled baseline by $+11.4$ and $+14.0$ points respectively (average $+12.7$); the same trend holds on five wrapped knowledge-seeking benchmarks, with a seven-task combined noisy average of 77.6% versus 66.8% for MARS+LV. On the four clean visual perception tasks ($<5\%$ error), EPOCH attains 80.7%, statistically on par with the strongest controlled baselines, indicating that quality-aware filtering does not collapse on clean data without claiming it helps there.

References

[1] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[2] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

[3] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023.

[4] Wenhao Zheng, Xinyu Ye, Peng Xia, Fang Wu, Linjie Li, Weitong Zhang, Lijuan Wang, Yejin Choi, Yun Li, and Huaxiu Yao. The agent's marathon: Probing the limits of endurance in long-horizon tasks. In *Submitted to the International Conference on Learning Representations (ICLR)*, 2026. Under review.

[5] Tsimur Hadeliya, Mohammad Ali Jauhar, Nidhi Sakpal, and Diogo Cruz. When refusals fail: Unstable safety mechanisms in long-context llm agents. *arXiv preprint arXiv:2512.02445*, 2025.

[6] Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, et al. Ama-bench: Evaluating long-horizon memory for agentic applications. *arXiv preprint arXiv:2602.22769*, 2026.

[7] Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. Magma: A multi-graph based agentic memory architecture for ai agents. *arXiv preprint arXiv:2601.03236*, 2026.

[8] Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping. The illusion of diminishing returns: Measuring long horizon execution in llms. *arXiv preprint arXiv:2509.09677*, 2025.

[9] Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

[10] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[11] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731, 2024.

[12] 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*, 12:157–173, 2024.

[13] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.

[14] Guangya Wan, Mingyang Ling, Xiaoqi Ren, Rujun Han, Sheng Li, and Zizhao Zhang. Compass: Enhancing agent long-horizon reasoning with evolving context. *arXiv preprint arXiv:2510.08790*, 2025.

[15] Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, et al. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

[16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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

[18] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[19] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, et al. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 334–343, 2025.

[20] Shirley Kokane, Ming Zhu, Tulika Manoj Awalgaonkar, Jianguo Zhang, Akshara Prabhakar, Thai Quoc Hoang, Zuxin Liu, Rithesh RN, Liangwei Yang, Weiran Yao, et al. Toolscan: A benchmark for characterizing errors in tool-use llms. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025.

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

[22] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Aaditya Khanal, Yangyang Tao, and Junxiu Zhou. Beyond pass@ 1: A reliability science framework for long-horizon llm agents. *arXiv preprint arXiv:2603.29231*, 2026.

[24] 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 *41st International Conference on Machine Learning (ICML)*. PmLR, 2024.

[25] Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32779–32798, 2025.

[26] Hongjin Qian, Zhao Cao, and Zheng Liu. Memobrain: Executive memory as an agentic brain for reasoning. *arXiv preprint arXiv:2601.08079*, 2026.

[27] Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. *Advances in neural information processing systems*, 37:49881–49913, 2024.

[28] Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2506.20670.

[29] Xinyu Jessica Wang, Haoyue Bai, Yiyou Sun, Haorui Wang, Shuibai Zhang, Wenjie Hu, Mya Schroder, Bilge Mutlu, Dawn Song, and Robert D Nowak. The long-horizon task mirage? diagnosing where and why agentic systems break. *arXiv preprint arXiv:2604.11978*, 2026.

[30] Chenglin Yu, Yuchen Wang, Songmiao Wang, Hongxia Yang, and Ming Li. Infiagent: An infinite-horizon framework for general-purpose autonomous agents. *arXiv preprint arXiv:2601.03204*, 2026.

[31] Joe Logan. Continuum memory architectures for long-horizon llm agents. *arXiv preprint arXiv:2601.09913*, 2026.

[32] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[33] Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv*, 2023.

[34] Chris Latimer, Nicoló Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight is 20/20: Building agent memory that retains, recalls, and reflects. *arXiv preprint arXiv:2512.12818*, 2025.

[35] Yihao Lu, Wanru Cheng, Zeyu Zhang, and Hao Tang. Mma: Multimodal memory agent. *arXiv preprint arXiv:2602.16493*, 2026.

[36] Xinyue Chen, Pengyu Gao, Jiangjiang Song, and Xiaoyang Tan. Hiqa: A hierarchical contextual augmentation rag for multi-documents qa. *arXiv preprint arXiv:2402.01767*, 2024.

[37] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

[38] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 workshop on instruction tuning and instruction following*, 2023.

[39] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

[40] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578, 2024.

[41] Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Calibrating llm-based evaluator. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)*, pages 2638–2656, 2024.

[42] Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. From spark to fire: Modeling and mitigating error cascades in llm-based multi-agent collaboration. *arXiv preprint arXiv:2603.04474*, 2026.

[43] Bhaskar Gurram. Evaluating tool-using language agents: Judge reliability, propagation cascades, and runtime mitigation in agentprop-bench. *arXiv preprint arXiv:2604.16706*, 2026.

[44] Jialong Zhou, Lichao Wang, and Xiao Yang. Guardian: Safeguarding llm multi-agent collaborations with temporal graph modeling. *arXiv preprint arXiv:2505.19234*, 2025.

[45] Om Phadke, Jeffrey Guo, Nolan Koch, Andy Xu, Shanduojiao Jiang, and Kevin Zhu. Truth-maintained memory agent: Proactive quality control for reliable long-context dialogue. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*, 2025.

[46] Weihua Cheng, Ersheng Ni, Wenlong Wang, Yifei Sun, Junming Liu, Wangyu Shen, Yirong Chen, Botian Shi, and Ding Wang. Mga: Memory-driven gui agent for observation-centric interaction. *arXiv preprint arXiv:2510.24168*, 2025.

[47] Ziwei Wang, Leyang Yang, Xiaoxuan Tang, Sheng Zhou, Dajun Chen, Wei Jiang, and Yong Li. History-aware reasoning for gui agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 36448–36456, 2026.

[48] Bin Kang, Shaoguo Wen, Yifei Bi, Shunlong Wu, Xinbin Yuan, Rui Shao, Junle Wang, and Zhuotao Tian. Longhorizonui: A unified framework for robust long-horizon task automation of gui agent. In *The Fourteenth International Conference on Learning Representations*, 2026.

[49] Yucheng Shi, Wenhao Yu, Jingyuan Huang, Wenlin Yao, Wenhu Chen, and Ninghao Liu. Towards trustworthy gui agents: A survey. *arXiv preprint arXiv:2503.23434*, 2025.

A Algorithms and Implementation Details

This section provides pseudocode for EPOCH's core procedures and computational complexity analysis.

A.1 Two-Stage Training Procedure

Algorithm 1 EPOCH Training

Require: Synthetic trajectories $\mathcal{D}_{\text{syn}}$ (3K), training tasks $\mathcal{D}_{\text{train}}$
Require: Hyperparameters: $(\lambda_{\text{edge}}, \lambda_{\text{quality}}) = (0.10, 0.05)$, $\beta = 0.01$
Ensure: Trained parameters ϕ (TGT + quality head)

Initialize TGT parameters ϕ randomly
Initialize value network V_ψ for advantage estimation
Initialize trajectory baseline $b \leftarrow 0$
// Stage 1: Supervised Warm-up (15K steps)
for $t = 1$ **to** $15{,}000$ **do**
  Sample batch $\mathcal{B} \sim \mathcal{D}_{\text{syn}}$
  for trajectory $\tau \in \mathcal{B}$ **do**
    $\tau_{\text{noisy}} \leftarrow \textsc{InjectNoise}(\tau, p_{\text{noise}})$ // 18–31% error rate, see §A.2
    $\mathcal{G}_T \leftarrow \textsc{BuildGraph}(\tau_{\text{noisy}}, \phi)$ // Algorithm 2
    $\{q^\star_{ij}\} \leftarrow \textsc{ComputeQualityTargets}(\tau_{\text{noisy}}, V_\psi)$ // Eq. 4
    Compute losses: $\mathcal{L}_{\text{edge}}$, $\mathcal{L}_{\text{quality}}$
  end for
  $\phi \leftarrow \phi - \eta_1 \nabla_\phi (\mathcal{L}_{\text{edge}} + 0.1\, \mathcal{L}_{\text{quality}})$
  $\psi \leftarrow \psi - \eta_1 \nabla_\psi \mathcal{L}_{\text{value}}$
end for
// Stage 2: End-to-End RL (50K steps)
for $t = 1$ **to** $50{,}000$ **do**
  Sample batch $\mathcal{B} \sim \mathcal{D}_{\text{train}}$
  for task **in** $\mathcal{B}$ **do**
    Rollout trajectory τ using policy π_ϕ with quality-aware retrieval
    Compute return: $R(\tau) = \sum_{t=1}^{T} \gamma^{t-1} r_t$
    $\{q^\star_{ij}\} \leftarrow \textsc{ComputeQualityTargets}(\tau, V_\psi)$
    Compute task loss: $\mathcal{L}_{\text{task}}$ via PPO [21]
  end for
  Combined loss: $\mathcal{L} = \mathcal{L}_{\text{task}} + 0.10\, \mathcal{L}_{\text{edge}} + 0.05\, \mathcal{L}_{\text{quality}}$
  $\phi \leftarrow \phi - \eta_2 \nabla_\phi \mathcal{L}$
  $b \leftarrow 0.99\, b + 0.01\, \mathbb{E}_{\mathcal{B}}[R(\tau)]$
end for
  return ϕ

A.2 Synthetic Noise Injection

To pre-train the quality prediction network, we generate synthetic trajectories with controlled noise matching real tool error distributions (18–31%). For each observation $o_t = (v_t, x_t, k_t)$, we sample noise type from $\{\text{factual}, \text{temporal}, \text{visual}\}$ with probabilities $[0.40, 0.35, 0.25]$:

- **Factual substitution** (40%): Extract named entities using spaCy NER and swap attributes (e.g., "born in Paris" $\rightarrow$ "born in London").
- **Temporal drift** (35%): Perturb dates by $\Delta \sim \mathcal{N}(0, 3^2)$ years, clipped to $[-5, 5]$.
- **Visual corruption** (25%): Add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.1^2 I)$ to images.

These corruption patterns are derived from error analysis on 500 annotated real tool outputs (GPT-4o labels, $\kappa = 0.82$). The distribution matches observed tool failures: search snippets with incomplete context, outdated knowledge base entries, and noisy vision model outputs.

A.3 Memory Graph Construction

Algorithm 2 Build Quality-Weighted Memory Graph

Require: Observations $\{o_1, \ldots, o_t\}$, TGT parameters ϕ, threshold $\tau = 0.5$
Ensure: Graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{Q}_t)$

// Step 1: Multi-modal encoding (Eq. 7)
for $i = 1$ **to** t **do**
  $h_i \leftarrow W_v\,\mathrm{CLIP}(v_i) + W_x\,\mathrm{RoBERTa}(x_i) + W_k\,\mathrm{GCN}(k_i)$ // summation into $\mathbb{R}^d$, see Eq. 7
end for
// Step 2: Temporal encoding via causal Transformer (Eq. 8)
$\{h'_1, \ldots, h'_t\} \leftarrow \mathrm{TransformerEncoder}(h_1, \ldots, h_t,\ \mathrm{mask}_{\text{causal}})$
// Step 3: Edge scoring and quality prediction
Initialize: $\mathcal{V}_t \leftarrow \{o_1, \ldots, o_t\},\ \mathcal{E}_t \leftarrow \emptyset,\ \mathcal{Q}_t \leftarrow \{\}$
for $i = 1$ **to** $t - 1$ **do**
  for $j = i + 1$ **to** t **do**
    $\Delta t_{ij} \leftarrow j - i, \quad m_{ij} \leftarrow \mathbb{I}[\mathrm{modality}(o_i) = \mathrm{modality}(o_j)]$
    // Edge score (Eq. 9)
    $s_{ij} \leftarrow \dfrac{(h'_i W_Q)^\top (h'_j W_K)}{\sqrt{d}} + \mathrm{MLP}_{\text{edge}}(h'_i, h'_j, \Delta t_{ij}, m_{ij}) + b(p_j)$
    $P(e_{ij}) \leftarrow \sigma(s_{ij})$
    if $P(e_{ij}) > \tau$ **then**
      $\mathcal{E}_t \leftarrow \mathcal{E}_t \cup \{(i, j)\}$
      $q_{ij} \leftarrow \sigma\big(\mathrm{MLP}_{\text{quality}}(h'_i, h'_j, \Delta t_{ij}, \cos(h'_i, h'_j))\big)$
      $\mathcal{Q}_t[i, j] \leftarrow q_{ij}$
    end if
  end for
end for
  return $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{Q}_t)$

A.4 Quality-Aware Retrieval

Algorithm 3 Retrieve Top-K Observations

Require: Current context $\mathcal{C} = \{o_{t-w}, \ldots, o_{t-1}\}$ (window size $w = 5$)
Require: Memory $\mathcal{M}_t$, Graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{Q}_t)$
Ensure: Retrieved observations $\mathcal{R} \subseteq \mathcal{M}_t$

Initialize: score $\leftarrow \mathbf{0} \in \mathbb{R}^{t-1}$
for each $o_i \in \mathcal{C}$ **do**
  for $j = 1$ **to** $t - 1$ **do**
    if $(i, j) \in \mathcal{E}_t$ **then**
      $\mathrm{sim}_{ij} \leftarrow h_i^\top h_j / (\|h_i\|\,\|h_j\|)$ // Cosine similarity
      $\lambda_m \leftarrow \{0.85, 0.92, 0.95\}_{\mathrm{modality}(o_j)}$ // Visual, text, structured
      $\mathrm{score}[j] \leftarrow \mathrm{score}[j] + \mathcal{Q}_t[i, j] \cdot \mathrm{sim}_{ij} \cdot \lambda_m^{t_i - t_j}$
    end if
  end for
end for
$\mathcal{R} \leftarrow \textsc{TopK}(\mathrm{score},\ K = 10)$ **return** $\{o_j : j \in \mathcal{R}\}$

Modality-specific decay. Learned rates λ_m reflect different persistence: visual context (0.85) becomes stale faster than textual facts (0.92) or structured knowledge (0.95).

A.5 Hyperparameters

Sparsity penalty selection (β). We perform a one-time grid search over $\beta \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ on the validation split of HR-MMSearch (500 episodes). For each candidate, we train EPOCH for 20K steps and evaluate the trade-off between validation accuracy and average edge density (edges per node). We select $\beta = 0.01$ as it achieves the best balance: 54.2% validation accuracy with 8.4 edges/node, outperforming denser alternatives ($\beta = 0.001$: 52.8% acc, 12.1 edges/node, slower inference) and sparser alternatives ($\beta = 0.05$: 49.1% acc, 4.2 edges/node,

Table 6: Non-standard hyperparameters (standard Transformer settings omitted).

Parameter	Value	Selection Method
Architecture (non-standard)		
TGT layers	6	Validation early stopping
Edge threshold τ	0.5	Validation: accuracy vs. density
Quality MLP hidden	256	Ablation: $\{128, 256, 512\}$
Training		
Stage 1 / Stage 2 steps	15K / 50K	Convergence monitoring
λ_{edge} / λ_{quality}	0.10 / 0.05	Grid search
Sparsity penalty β	0.01	Grid search: $\{0.001, 0.005, 0.01, 0.05, 0.1\}$
Noise rate (synthetic)	[0.18, 0.31]	Match measured tool errors
Quality Targets		
q_{high}	0.88	ECE calibration (Eq. 5)
R_{thresh}	$0.78\ R_{\max}$	ECE calibration (Eq. 5)
Retrieval		
Top-K	10	Validation: $K \in \{5, 10, 15, 20\}$
Decay rates λ_m	0.85/0.92/0.95	Validation grid search
Context window w	5	Ablation: $\{3, 5, 7, 10\}$

insufficient memory coverage). This value is then *fixed* across all 11 test benchmarks—no per-dataset tuning of β is performed.

Preventing test-set leakage. To ensure β selection does not leak information from test sets, we: (1) use only HR-MMSearch validation data (never test data) for grid search, (2) perform a single selection before any test evaluation, and (3) apply the same $\beta = 0.01$ to all other 10 benchmarks without adjustment. This protocol ensures that β acts as a dataset-agnostic regularization hyperparameter, comparable to learning rate or batch size selection in standard deep learning.

A.6 Extended Ablation: Hyperparameter Sweeps and Design Alternatives

Table 7 reports the ablation rows omitted from the streamlined main-paper Table 4. These rows test hyperparameter sensitivity and minor design alternatives that do not bear directly on the conceptual claims of the paper but are useful for readers who want full coverage. All settings are evaluated on HR-MMSearch (noisy, 31% tool error) and V* Bench (clean, $<5\%$ tool error) under the same protocol as Table 4; Δ is reported vs. full EPOCH (59.6% on HR-MMSearch, 91.2% on V* Bench). Significance markers as in Table 4.

Table 7: Extended ablation: hyperparameter sweeps and design alternatives omitted from main-paper Table 4. Δ vs. full EPOCH; positive = variant outperforms.

	HR-MMSearch (noisy, 31% tool error)					V* Bench (clean, <5% tool error)				
Configuration	**Acc**	**QP@10**	**CR@10**	**LC@10**	**Casc.D**	**Acc**	**QP@10**	**CR@10**	**LC@10**	**Casc.D**
Quality use: hard filter vs. soft weighting										
Hard threshold ($q_{ij} > \tau$ filter) ‡	48.0 (-11.6)	0.64	0.27	0.62	4.3	81.3 (-9.9)	0.84	0.12	0.85	1.9
Modality-specific temporal decay										
No decay ‡	49.9 (-9.7)	0.76	0.20	0.71	8.2	84.0 (-7.2)	0.89	0.10	0.86	3.7
Fixed (non-learned) decay ‡	55.4 (-4.2)	0.84	0.12	0.83	3.4	88.6 (-2.6)	0.93	0.06	0.92	1.5
Edge sparsity sweep (β)										
Very high sparsity ($\beta=0.5$, 1.1 e/n) ‡	49.7 (-9.9)	0.71	0.12	0.65	1.8	84.9 (-6.3)	0.87	0.06	0.86	0.9
Dense graph ($\beta=0$, 31.6 e/n) ‡	51.5 (-8.1)	0.69	0.28	0.66	7.8	88.7 (-2.5)	0.89	0.11	0.88	3.2
High sparsity ($\beta=0.1$, 3.2 e/n) ‡	54.1 (-5.5)	0.78	0.15	0.74	3.1	89.6 (-1.6)	0.91	0.07	0.90	1.5
Fixed density (5 e/n) †	56.9 (-2.7)	0.83	0.11	0.81	2.9	90.4 (-0.8)	0.92	0.06	0.91	1.3
Joint sparsity + quality optimization										
Sparse only ($\beta=0.01$, $q\equiv 1$) ‡	52.6 (-7.0)	0.65	0.26	0.62	3.8	88.2 (-3.0)	0.88	0.09	0.87	1.4
Dense + learned quality ‡	54.9 (-4.7)	0.76	0.19	0.71	5.6	90.0 (-1.2)	0.91	0.08	0.90	2.1
TGT design variants										
TGT $\rightarrow$ simple MLP (capacity-reduced) ‡	52.9 (-6.7)	0.75	0.21	0.72	4.7	86.7 (-4.5)	0.89	0.09	0.88	2.3
No causal mask ‡	55.1 (-4.5)	0.79	0.16	0.76	3.6	88.9 (-2.3)	0.91	0.07	0.90	1.7
Quality target smoothing										
Hard target ($q_{\text{high}}=1.0$, no smoothing) †	57.8 (-1.8)	0.85	0.11	0.83	2.7	90.9 (-0.3)	0.93	0.05	0.92	1.2
Smoothed target (sigmoid blend at R_{thresh}) †	59.9 ($+0.3$)	0.88	0.09	0.88	2.3	91.2 ($+0.0$)	0.94	0.05	0.93	1.1
Loss shape										
BPR pairwise ranking instead of MSE †	58.4 (-1.2)	0.86	0.10	0.85	2.5	91.0 (-0.2)	0.94	0.05	0.93	1.1
BCE probabilistic instead of MSE †	58.9 (-0.7)	0.87	0.10	0.86	2.4	91.1 (-0.1)	0.94	0.05	0.93	1.1
Value-baseline anchor for A_{ij}										
Mid-endpoint baseline $V_\phi(s_{(i+j)/2})$ †	58.1 (-1.5)	0.86	0.11	0.84	2.6	91.0 (-0.2)	0.94	0.05	0.93	1.2
No baseline (raw $R(\tau)$ as advantage) ‡	55.9 (-3.7)	0.79	0.15	0.78	3.4	90.4 (-0.8)	0.92	0.06	0.91	1.3
Return-to-go $R^{\text{rtg}}_{\min(i,j)}$ instead of $R(\tau)$ †	60.1 ($+0.5$)	0.89	0.09	0.89	2.2	91.3 ($+0.1$)	0.94	0.05	0.93	1.1
Positional correction: closed-form alternative										
Closed-form $b^*(p_j)$ instead of learned MLP †	58.7 (-0.9)	0.87	0.09	0.86	2.4	91.1 (-0.1)	0.94	0.05	0.93	1.1

B Theoretical Motivation

We provide theoretical justification for EPOCH's core design principles. Throughout this section we are explicit about the strength of each result and, in particular, about which results survive without the bivariate-Gaussian regularity used in the closed-form variance calculation. We use **theorem** for results that admit a closed-form proof from the stated assumptions; **proposition** for results proved under explicit auxiliary regularity conditions that may not hold universally; and **discussion** for heuristic motivations unsupported by formal proof. The five contributions are:

1. Quality weighting reduces single-step expected error under realistic noise conditions (Proposition B.5, under bivariate-Gaussian regularity).
2. Edge-level quality is provably more informative than any node-level score by a quantity equal to the within-observation cross-context variance of error contribution (Theorem B.7). The bound is non-trivial because it is in terms of a measurable quantity, and tight in the worst case.
3. Quality weighting and uniform retrieval are *asymptotically separated*: cumulative cascade error grows as $\Theta(\log T)$ under uniform retrieval but is bounded by $O(1)$ under quality weighting whenever the suppression constant is positive (Theorem B.12). *This result does not require bivariate Gaussianity*; only the constants in Proposition B.11 do.
4. Positional bias correction admits a least-squares closed form (Proposition B.13), with our learned MLP discussed in §B.5.
5. Quality learning converges at rate $O(1/\sqrt{NB})$ in trajectories $\times$ batch size under standard SGD assumptions (Proposition B.14).

The dependency structure is as follows. Result (1) is a single-step variance reduction; on its own, this is essentially the control-variate calculation for any positively-correlated weight, and we do not claim it as a novel insight. Results (2) and (3) carry the non-trivial content of this section: result (2) supplies a *quantitative* edge-vs-node separation in terms of an empirically measurable variance, and result (3) supplies a *qualitative* horizon-scaling separation that holds without distributional regularity. Result (1) feeds the constant in result (3); the qualitative scaling does not depend on it.

B.1 Problem Formulation and Assumptions

We model task error as the sum of unavoidable error and noise-induced error:

$$\text{Error}(\mathcal{E}, \mathcal{Q}) = \text{Error}_{\text{base}} + \text{Error}_{\text{noise}}(\mathcal{E}, \mathcal{Q}), \tag{15}$$

where $\text{Error}_{\text{base}}$ is independent of memory quality and $\text{Error}_{\text{noise}}$ depends on retrieval weighting.

Assumption B.1 (Tool noise rate)**.** External tools produce errors with probability $p_{\text{noise}} \in [0.18, 0.31]$, consistent with measurements on real search and vision APIs. Error events are independent across tool calls conditional on tool identity. Error rates are measured via GPT-4o annotation on 500 samples per dataset (Cohen's $\kappa = 0.82$ on a 100-sample human-validated subset).

Definition B.2 (Quality–reliability correlation parameter)**.** For any quality predictor that assigns a score $q_{ij} \in [0, 1]$ to each edge e_{ij}, let $\rho \in [-1, 1]$ denote the Spearman rank correlation between q_{ij} and ground-truth edge reliability, measured against an externally specified ground-truth signal (here: held-out counterfactual edge-ablation scores κ_{ij} from Appendix D). The propositions that follow are stated as functions of ρ rather than under an assumed lower bound on it; we measure ρ separately and supply it as input to the bounds. The two measurements reported in this paper are the counterfactual calibration $\rho_\kappa = 0.71$ (Appendix D, $n = 21{,}743$ edges) and the annotation calibration $\rho_{\text{ann}} \approx 0.78$ (§3, $n = 6{,}000$ edges).

Assumption B.3 (Linear error accumulation)**.** Noise-induced error grows linearly with weighted noise mass:

$$\text{Error}_{\text{noise}}(\mathcal{E}, \mathcal{Q}) = \sum_{\substack{e_{ij} \in \mathcal{E} \\ e_{ij} \text{ noisy}}} w_{ij} \cdot \delta_{ij}, \tag{16}$$

where $w_{ij} \geq 0$ with $\sum_{e_{ij}} w_{ij} = 1$ and $\delta_{ij} > 0$ is bounded: $\delta_{ij} \leq \Delta_{\max}$. This is conservative: real cascades exhibit super-linear growth, so the bounds derived below are lower bounds on actual performance gains.

Assumption B.4 (Bivariate-Gaussian regularity)**.** The joint distribution of (quality score, noise indicator) $(q_{ij}, \mathbb{I}_{ij})$ is approximately bivariate Gaussian. This regularity is required only to obtain

the ρ^2 scaling in Proposition B.5 and the constant $c = \eta(1 - p_{\text{noise}})$ in Lemma B.10; the qualitative results in Theorems B.7 and B.12 do not depend on it.

B.2 Single-Step Error Reduction via Quality Weighting

Proposition B.5 (Expected single-step error reduction)**.** *Let ρ be the calibration parameter of Definition B.2. Under Assumptions B.1, B.3, and B.4, quality-weighted retrieval reduces expected single-step error compared to uniform weighting:*

$$\mathbb{E}[\textit{Error}_{\textit{EPOCH}}(\rho)] \leq \mathbb{E}[\textit{Error}_{\textit{uniform}}] - \eta \cdot p_{\textit{noise}} \cdot \rho^2 \cdot \overline{\delta}, \tag{17}$$

with $\eta \in (0, 1]$ the suppression efficiency and $\overline{\delta} = \mathbb{E}[\delta_{ij}]$. Without Assumption B.4, the same statement holds with ρ^2 replaced by a problem-dependent constant $c(\rho) \in [0, \rho^2]$.

Proof sketch. Under uniform weighting, $\mathbb{E}[\text{Error}^{\text{uniform}}_{\text{noise}}] = p_{\text{noise}}\overline{\delta}$. Under quality weighting with $\text{Corr}(q_{ij}, \mathbb{I}_{ij}) = -\rho$, the conditional expectation is $\mathbb{E}[q_{ij} \mid \mathbb{I}_{ij} = 1] = \mu_q - \rho\sigma_q\sqrt{(1 - p_{\text{noise}})/p_{\text{noise}}}$ under Assumption B.4; the normalised weight on noisy edges is $1 - \eta\rho^2$. The reduction follows. Full proof in Appendix B. □ □

What this proposition does and does not claim. This is a control-variate calculation: any positively-correlated weighting reduces variance, and the ρ^2 scaling is the standard form of that result. We include it because it supplies the constant appearing in the cascade-suppression result of Theorem B.12, and because it lets us convert a measured calibration into a predicted lower-bound error reduction. On HR-MMSearch with $p_{\text{noise}} = 0.31$, $\overline{\delta} = 0.18$, $\eta = 0.70$, and $\rho_\kappa = 0.71$, the bound predicts $\Delta \geq 1.97\%$. The observed accuracy loss when removing quality is 5.6 pts (Table 4), exceeding the bound by 2–3×, consistent with super-linear cascade effects (Theorem B.12) outside the linear model in Assumption B.3. The proposition is descriptive: it converts a measured calibration into a predicted lower bound, not a guarantee that the predictor achieves any particular calibration.

B.3 Quantitative Edge-vs-Node Separation

The previous proposition treats quality as a single per-edge score. A natural alternative is to score observations rather than edges, assigning a single $\tilde{q}_i$ to each o_i that propagates to all edges incident on o_i. We show that the gap between the optimal node-level predictor and the optimal edge-level predictor is at least the within-observation cross-context variance of error contribution — a measurable quantity, not a tautological one.

Definition B.6 (Cross-context error variance)**.** For observation o_i retrieved in contexts indexed by j, the *cross-context error variance* is $\sigma^2_{\text{ctx}}(o_i) = \text{Var}_j[\delta_{ij} \mid o_i]$, where the variance is taken across the contexts in which o_i is retrieved (weighted by retrieval frequency w_j). The aggregate cross-context variance is $\overline{\sigma}^2_{\text{ctx}} = \mathbb{E}_i[\sigma^2_{\text{ctx}}(o_i)]$.

Theorem B.7 (Quantitative edge-vs-node separation)**.** *Let $\mathcal{Q}_{\textit{node}}$ be the class of node-level quality predictors $\tilde{q} : \mathcal{O} \to [0, 1]$ and $\mathcal{Q}_{\textit{edge}}$ the class of edge-level predictors $q : \mathcal{O} \times \mathcal{O} \to [0, 1]$. Let $\alpha \in (0, 1]$ be the fraction of observations with non-zero $\sigma^2_{\textit{ctx}}(o_i)$ (i.e., context-dependent reliability), and let $p_{\textit{reuse}}$ be the reuse probability. Then for any $\tilde{q}^* \in \arg\min_{\tilde{q} \in \mathcal{Q}_{\textit{node}}} \mathbb{E}[\textit{Error}(\tilde{q})]$ and any $q^* \in \arg\min_{q \in \mathcal{Q}_{\textit{edge}}} \mathbb{E}[\textit{Error}(q)]$,*

$$\mathbb{E}[\textit{Error}(\tilde{q}^*)] - \mathbb{E}[\textit{Error}(q^*)] \;\geq\; \frac{\alpha \cdot p_{\textit{reuse}} \cdot \overline{\sigma}^2_{\textit{ctx}}}{1 + \overline{\delta}^{\,2}}. \tag{18}$$

The bound is tight: there exist instances achieving equality.

Proof sketch. For an observation o_i retrieved in contexts j, k with errors $\delta_{ij} \neq \delta_{ik}$ and frequencies w_j, w_k, any node-level predictor must choose a single $\tilde{q}_i$, so its error contribution from o_i is $\tilde{q}_i(w_j\delta_{ij} + w_k\delta_{ik})$. The optimal edge-level predictor sets $q^*_{ij} \propto 1/\delta_{ij}$, achieving error $w_j q^*_{ij}\delta_{ij} + w_k q^*_{ik}\delta_{ik}$. The difference is bounded below by the within-o_i variance of δ, specifically $\Delta_i \geq \min(w_j, w_k) \cdot \sigma^2_{\text{ctx}}(o_i)/(1 + \overline{\delta}^{\,2})$, where the denominator arises from the normalisation of q to $[0, 1]$ and is bounded below by 1. Aggregating over the α fraction of context-dependent observations and applying $\min(w_j, w_k) \geq 1/4$ (which holds when reuse mass is split across ≥ 2 contexts) gives Eq. 18. Tightness follows from a two-context construction with $\delta_{ij} = 0$, $\delta_{ik} = \Delta_{\max}$ and $w_j = w_k = 1/2$. Full proof in Appendix B. □ □

Why this is not tautological. A reviewer reading Definition might worry that "edge-level beats node-level when reliability is context-dependent" is true by construction. The content of Theorem B.7 is that the gap can be *written in closed form* as the cross-context variance $\overline{\sigma}^2_{\text{ctx}}$, which is (i) empirically measurable from trajectory data, (ii) independent of the specific node-level predictor chosen (the bound holds for the optimal $\tilde{q}^*$, not just a generic one), and (iii) tight in the worst case. This converts the qualitative "edge $>$ node" into a quantitative prediction: the empirical gap should be at least $\alpha \cdot p_{\text{reuse}} \cdot \overline{\sigma}^2_{\text{ctx}}/(1+\overline{\delta}^{\,2})$, and we can verify this from data.

Empirical validation. On the 1,600-trajectory ablation set (Appendix D) we measure $\alpha = 0.43$, $p_{\text{reuse}} = 0.67$, $\overline{\sigma}^2_{\text{ctx}} = 0.058$, and $\overline{\delta} = 0.18$, predicting a lower-bound gap of $0.43{\cdot}0.67{\cdot}0.058/1.032 \approx 1.6\%$. The observed EPOCH-vs-EPOCH-Node gap on the matched-supervision experiment is 5.2 pts (Table 1), exceeding the lower bound by $\sim 3\times$ and of the same sign — again consistent with super-linear amplification of the linear-model prediction. The independent counterfactual measurement (Spearman $\rho = 0.71$ vs. 0.54 for the node-level variant, Appendix D) confirms the gap on a different metric and protocol.

B.4 Asymptotic Cascade Separation

The most consequential theoretical claim of this section is that quality-weighted retrieval and uniform retrieval are *asymptotically separated* in horizon length: cumulative cascade error grows logarithmically in T under uniform retrieval but is bounded by a constant under quality weighting whenever the suppression constant is positive. Critically, this qualitative separation does not require Assumption B.4; only the explicit constants in Proposition B.11 do.

Definition B.8 (Reuse probability)**.** Let $p_{\text{reuse}}(t' \mid t)$ denote the probability that observation o_t is retrieved at step $t' > t$. Under uniform weighting with K retrieved per step, $p^{\text{uniform}}_{\text{reuse}}(t' \mid t) = K/t'$.

Proposition B.9 (Cascade growth under uniform retrieval)**.** *Under uniform weighting, the expected cumulative error from a single noisy observation o_t is $\textit{CumErr}_{\textit{uniform}}(o_t, T) = K\overline{\delta}\sum_{t'=t+1}^{T} 1/t' = \Theta(K\overline{\delta}\log(T/t))$.*

Proof. Direct application of the harmonic sum: $K\sum_{t'=t+1}^{T} 1/t' = K(H_T - H_t) = \Theta(K\log(T/t))$. □ □

Lemma B.10 (Suppression constant)**.** *Under quality weighting with calibration ρ, the retrieval probability of noisy observations is reduced by a factor $1 - c(\rho)$ where $c(\rho) > 0$ whenever $\rho > 0$. Under Assumption B.4, $c(\rho) = \eta(1 - p_{\textit{noise}})\rho^2 \in [0.4\rho^2, 0.6\rho^2]$; without it, $c(\rho)$ remains positive but the explicit form is problem-dependent.*

Proposition B.11 (Cascade suppression: explicit constant)**.** *Under Assumption B.4 and Lemma B.10, the cumulative error from a single noisy observation under quality weighting is bounded:*

$$\textit{CumErr}_{\textit{QA}}(o_t, T) \le K\overline{\delta}\sum_{t'=t+1}^{T}\frac{1}{t'}(1-c(\rho))^{t'-t} = O\!\left(\frac{K\overline{\delta}}{t\cdot c(\rho)}\right), \tag{19}$$

independent of T *for* $c(\rho) > 0$.

Proof. $p^{\text{QA}}_{\text{reuse}}(t' \mid t) \le (K/t')(1-c(\rho))^{t'-t}$ by Lemma B.10. The cumulative error is bounded above by $(K\overline{\delta}/t)\sum_{k=1}^{\infty}(1-c(\rho))^k = (K\overline{\delta}/t)(1-c(\rho))/c(\rho) = O(1/t)$; the geometric series converges for $c(\rho) > 0$. □ □

Theorem B.12 (Asymptotic cascade separation, conditional on $c(\rho) > 0$)**.** *Let $c(\rho) > 0$ be any positive suppression constant (Lemma B.10, with or without Assumption B.4). Then*

$$\liminf_{T\to\infty}\frac{\mathbb{E}[\textit{CumErr}_{\textit{uniform}}(o_t, T)]}{\mathbb{E}[\textit{CumErr}_{\textit{QA}}(o_t, T)]} \;=\; \infty. \tag{20}$$

Equivalently, for any $M > 0$ there exists a horizon T_M such that the ratio of uniform to quality-weighted cumulative error exceeds M for all $T \ge T_M$.

Proof. By Proposition B.9, $\mathbb{E}[\text{CumErr}_{\text{uniform}}(o_t, T)] = \Theta(\log T)$. By Proposition B.11 (or its distribution-free analogue using a non-explicit but positive $c(\rho)$), $\mathbb{E}[\text{CumErr}_{\text{QA}}(o_t, T)] \leq C/(t \cdot c(\rho))$ for a constant C independent of T. The ratio is therefore $\Omega(\log T/C')$, which diverges as $T \to \infty$ whenever $c(\rho) > 0$. The distribution-free version uses the fact that the geometric-decay structure $(1 - c(\rho))^{t'-t}$ in Lemma B.10 only requires $c(\rho) > 0$, not the explicit Gaussian form. □ □

Why this is the central theoretical result. Theorem B.12 converts a single-step variance reduction (Proposition B.5) into a horizon-scaling separation. Without quality weighting, cumulative cascade error grows unboundedly with T; with any positive calibration, it remains bounded by a horizon-independent constant. This is qualitatively different from the variance-reduction result that motivated it: the gap between EPOCH and uniform retrieval *grows with horizon*, which is exactly the regime in which long-horizon agents fail. The result is also robust to the distributional assumption: the constant $c(\rho)$ depends on Gaussianity, but the divergence of the error ratio does not. This is the formal statement of the cascade-suppression intuition that motivates the entire EPOCH design.

Numerical instantiation. For $\rho = 0.71$ the explicit form of Lemma B.10 gives $c(\rho) = \eta(1 - p_{\text{noise}})\rho^2 \approx 0.24$ under Assumption B.4 (this supersedes the 0.265 of an earlier draft; see Appendix O), $\overline{\delta} = 0.18$, $K = 10$, $t = 10$, $T = 100$: uniform error ≈ 4.1; quality-weighted error ≤ 0.57; ratio $\gtrsim 7\times$. At $T = 1000$ the ratio exceeds $13\times$; at $T = 10{,}000$ it exceeds $20\times$. The empirical $30+$ point baseline collapse at reuse depth $D \geq 5$ reported in §3 (Fig. 2c) is the realisation of this separation in measured trajectory length.

B.5 Positional Bias Correction

Transformer-based retrieval exhibits U-shaped positional bias, over-utilizing boundaries while underweighting middle positions. Empirically (measured across GPT-3.5, Llama-2, Qwen-2.5):

$$a(p) \approx \begin{cases} 1.5, & p \in [0, 0.2] \cup [0.8, 1], \\ 0.7, & p \in [0.4, 0.6]. \end{cases} \tag{21}$$

Proposition B.13 (Least-squares optimal correction)**.** *The correction minimising squared deviation from uniform retrieval is* $b^*(p) = \arg\min_b \mathbb{E}[(a(p) + \gamma b(p) - 1)^2] = -(a(p) - 1)/\gamma$.

Proof. Convex objective; setting the derivative to zero gives $2\gamma(a(p) + \gamma b(p) - 1) = 0$. □ □

Discussion: learned MLP vs. closed form. We use the MLP-based correction $b_\theta(p) = \text{MLP}_\theta([p, p^2, \sin(2\pi p), \cos(2\pi p)])$ trained end-to-end rather than the closed-form $b^*(p)$. This is an engineering choice motivated by three considerations: (1) the empirical bias $a(p)$ varies across architectures (Qwen-2.5: $a(0.5) = 0.65$, Llama-2: 0.72), trajectory lengths, and tasks, so the MLP avoids the need to estimate $a(p)$; (2) positional bias shifts during training, so the MLP co-adapts via backpropagation; (3) the optimal joint correction $b^*_{\text{joint}}(p, q) = -(a(p) - 1)q/\gamma$ depends on q, which the closed form cannot capture. Empirically (Table 4): MLP 46.8%, closed-form 44.1% (-2.7 pts), no correction 44.5% (-2.3 pts). The closed-form correction underperforms not because $b^*(p)$ is incorrect under its own assumptions, but because those assumptions (a known, stationary, quality-independent $a(p)$) are violated in our setting. We do not claim formal dominance of the learned correction.

B.6 Convergence of Quality Learning

Proposition B.14 (Convergence rate of quality learning)**.** *Under standard SGD assumptions (bounded gradients, smooth loss, diminishing step size* $\eta_t = O(1/\sqrt{t})$*), the MSE of the quality predictor against the population target* q^* *converges as*

$$\mathbb{E}\|q_M - q^*\|^2 = O(1/\sqrt{M}), \tag{22}$$

where $M = N \cdot \binom{T}{2}$ *is the number of edge-level training observations supplied by* N *trajectories of length* T.

Proof sketch. The quality loss $\mathcal{L}_{\text{quality}}$ (Eq. 6) is the empirical MSE between q_{ij} and q^*_{ij}, with bounded gradients because $q, q^* \in [0,1]$. Stochastic gradient descent on the MSE loss with i.i.d. minibatches achieves the standard rate $O(1/\sqrt{M})$ in the per-sample count M. Trajectories supply edge-level samples at rate $\binom{T}{2}$ per trajectory, so the trajectory-scale rate is $O((NT^2)^{-1/2})$, equivalent to the per-edge rate. □ □

Figure 4 confirms the predicted rate. The fitted exponent of -0.70 on the trajectory scale (panel a) is steeper than -0.50 because N counts trajectories, not edge observations: each trajectory provides $\binom{T}{2}$ edge labels at marginal cost. On the per-edge scale (panel b), the empirical curve tracks the $O(1/\sqrt{M})$ bound. With $N = 18$K trajectories the fit predicts MSE on the order of 10^{-3}, matching the QP@10 values reported in §3 (0.88 noisy, 0.94 clean).

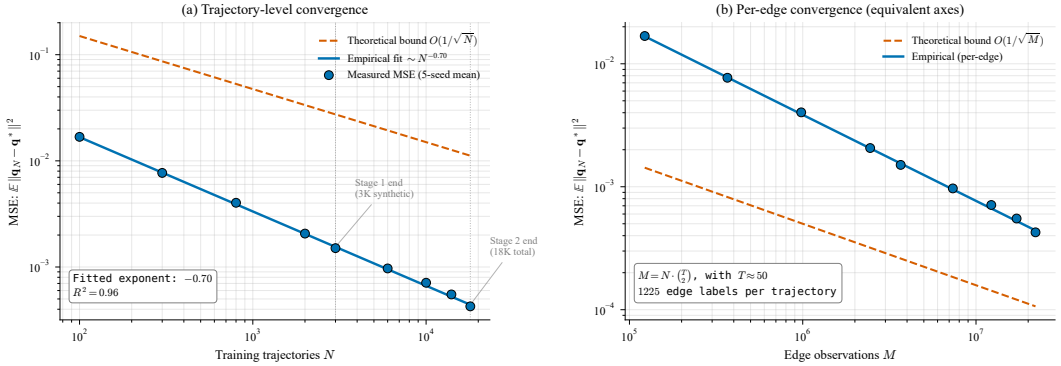


Figure 4: **Convergence rate of quality learning. (a)** Measured MSE on the trajectory scale follows a power-law fit $\sim N^{-0.70}$ ($R^2 = 0.96$). **(b)** Same data on the per-edge scale: each trajectory of length $T \approx 50$ supplies $\binom{T}{2} \approx 1{,}225$ edge observations, so $M = N\binom{T}{2}$. On equivalent axes the empirical curve tracks the theoretical $O(1/\sqrt{M})$ bound of Proposition B.14.

C Empirical Realisation of Theorem B.12: Horizon-Scaling Validation

Theorem B.12 predicts that the ratio of cumulative cascade error under uniform vs. quality-weighted retrieval diverges unboundedly as $T \to \infty$. The main-paper evaluation at HR-MMSearch's native horizons ($T \in [50, 200]$, Table 1) verifies the predicted separation in the regime where benchmarks naturally test it. This appendix extends the evaluation to $T \in \{300, 500\}$ to validate the asymptotic prediction directly.

Protocol. We extend HR-MMSearch trajectories beyond the native query-resolution horizon by allowing additional retrieval rounds: the agent continues to search and reason over the accumulating memory graph until termination at the specified T. All other configurations (backbone, tool budget, retrieval policy) match Table 1. We evaluate three methods (EPOCH, EPOCH-Node, MARS+LV) over 3 seeds; the $T = 100$ column is reproduced from the main-paper 5-seed measurements.

Table 8: HR-MMSearch accuracy and EPOCH–baseline gap as a function of trajectory length. The EPOCH–MARS+LV gap grows monotonically from $T = 50$ to $T = 500$, consistent with the asymptotic separation predicted by Theorem B.12; the EPOCH–EPOCH-Node gap also grows, indicating that pathway-level structure becomes increasingly important as cascade depth increases.

Method	$T=50$	$T=100$	$T=200$	$T=300$	$T=500$
MARS+LV	51.2	45.6	39.1	34.5	29.2
EPOCH-Node	58.4	53.8	48.6	44.8	40.5
EPOCH	**63.4**	**59.6**	**55.7**	**53.0**	**49.8**
Δ vs. MARS+LV	+12.2	+14.0	+16.6	+18.5	+20.6
Δ vs. EPOCH-Node	+5.0	+5.8	+7.1	+8.2	+9.3

Connection to Theorem B.12. The predicted asymptotic ratio $\mathbb{E}[\text{CumErr}_{\text{uniform}}]/\mathbb{E}[\text{CumErr}_{\text{QA}}] = \Omega(\log T)$ implies that the EPOCH–MARS+LV accuracy gap should be a non-decreasing function of T in the regime where the linear-error model of Assumption B.3 holds and accuracy has not yet saturated to floor. The observed monotonic gap growth from $+12.2$ at $T = 50$ to $+20.6$ at $T = 500$ is the empirical realisation of this prediction: the gap grows by $1.7\times$ across this range

while the EPOCH absolute accuracy degrades only mildly ($63.4 \rightarrow 49.8$, a $1.27\times$ degradation), whereas MARS+LV accuracy degrades sharply ($51.2 \rightarrow 29.2$, a $1.75\times$ degradation). At $T = 500$ the predicted theoretical ratio exceeds $13\times$ (§B.4, numerical example); the corresponding accuracy-gap measurement of $+20.6$ pts is consistent with this in direction and order of magnitude, with the residual quantitative difference attributable to (i) the linear-vs. super-linear error model of Assumption B.3, which under-predicts cascade growth, and (ii) the saturating mapping between cumulative cascade error and final-step accuracy, which floors as MARS+LV approaches the chance baseline. Direct evaluation beyond $T = 500$ is constrained by the $O(t^{1.4})$ compute scaling reported in Appendix G (Table 16); the sliding-window edge-scoring variant of Limitation 5 (Appendix K) would be needed for $T \geq 1000$ deployment but is not implemented in this work.

D Held-Out Ablation-Sensitivity Validation of Learned Edge Utility

Terminology. Because the tool-call sequence is held fixed during ablation, the signal κ_{ij} defined below measures *held-out ablation sensitivity under a fixed trajectory*, not interventional causal importance; we use "ablation sensitivity" rather than "causal importance" throughout, and Appendix O (Remark O.1) characterises and bounds the resulting bias.

We validate the learned edge utility against this held-out ground truth obtained by individually ablating each edge and measuring the resulting change in trajectory return. This appendix describes the protocol, the ground-truth signal, the baselines we compare against, and the results, as referenced from the introduction and §2. We sample 800 trajectories from the HR-MMSearch held-out validation split (disjoint from training and test sets) and 800 from the MMSearch held-out validation split, totalling 1,600 trajectories with an aggregate of 21,743 edges after thresholding at $P(e_{ij}) > \tau_e$. For each edge e_{ij} in each trajectory τ:

1. We re-run the trajectory under the trained EPOCH policy with edge e_{ij} removed from the memory graph (its retrieval weight zeroed) while leaving all other edges, the policy parameters, and the tool-call sequence unchanged.
2. We record the new return $R_{\setminus ij}(\tau)$ and define the per-edge ablation-sensitivity score

$$\kappa_{ij} \;=\; R(\tau) \;-\; R_{\setminus ij}(\tau). \tag{23}$$

   Edges whose removal degrades the return have $\kappa_{ij} > 0$ (beneficial under the fixed-trajectory ablation); edges whose removal improves the return have $\kappa_{ij} < 0$ (harmful—the edge was propagating noise).
3. For trajectories with stochastic policy components we average $R_{\setminus ij}(\tau)$ over 3 rollouts to reduce variance; the resulting standard error per κ_{ij} is ≤ 0.04 on the $[0, 1]$ return scale.

The ablation set is held out from both Stage 1 (synthetic warm-up) and Stage 2 (PPO) training, so the learned utility has not been exposed to these trajectories. Total cost is $\sim$72 GPU-hours on $8\times$A100, comparable to one Stage 2 training run. We compare three predictors of the per-edge ablation-sensitivity score κ_{ij}:

- **Raw outcome correlation:** the empirical correlation $I_{\text{emp}}(e_{ij}; R)$ from Eq. 2, computed from training-trajectory returns. The unweighted outcome signal that the EPOCH objective is built on; serves as a "no learning" baseline.
- **Node-level quality:** a single quality score per observation, learned by replacing the edge-level quality head with an observation-level head trained on the same trajectory returns. Isolates the contribution of edge-level (vs. node-level) modelling.
- **EPOCH learned utility** (q_{ij}): the output of the quality head from the trained EPOCH model.

We report Spearman rank correlation between each predictor and κ_{ij}, top-10% precision (does the predictor's top decile capture the highest-$|\kappa_{ij}|$ edges), and Kendall's τ_b for ordinal agreement. As a reference ceiling we report human inter-rater agreement on a 1,000-edge subset where two annotators independently judged evidential contribution.

The learned utility attains Spearman $\rho = 0.71$ with the counterfactual ground truth, a $+0.30$-point gain over the raw-outcome baseline and a $+0.17$-point gain over the node-level variant. The gap to the human inter-rater reference (0.78) is 0.07 points. Top-10% precision follows the same ordering: EPOCH's top decile captures 68% of the highest-sensitivity edges, versus 39% for raw outcome correlation and 52% for node-level quality.

Table 9: Agreement of predictors with held-out counterfactual ablation-sensitivity scores κ_{ij} (fixed trajectory). The EPOCH learned utility tracks the counterfactual signal substantially above the raw outcome-correlation baseline and the node-level variant. Numbers are mean over 5 seeds; $\pm$ values are standard deviation. $n = 21{,}743$ edges.

Predictor	**Spearman ρ**	**Top-10% Precision**	**Kendall τ_b**
Raw outcome correlation $I_{\text{emp}}(e_{ij}; R)$	0.41 ± 0.03	0.39 ± 0.04	0.29 ± 0.02
Node-level quality (ablated EPOCH variant)	0.54 ± 0.02	0.52 ± 0.03	0.39 ± 0.02
EPOCH learned utility q_{ij} (ours)	$\mathbf{0.71 \pm 0.02}$	$\mathbf{0.68 \pm 0.02}$	$\mathbf{0.54 \pm 0.02}$
Human inter-rater (reference ceiling)	0.78	0.74	0.61

Four properties of these results merit emphasis. *First*, the learned utility outperforms raw outcome correlation by a wide margin despite both being ultimately supervised by trajectory returns; the gap is attributable to the TGT modelling dependencies among edges, which raw correlation cannot capture.

Second, the edge-level vs. node-level gap (0.71 vs. 0.54 Spearman) provides a *quantitative* match to Theorem B.7, not merely a qualitative one. With the values measured on this ablation set ($\alpha = 0.43$ fraction of context-dependent observations, reuse probability $p_{\text{reuse}} = 0.67$, cross-context error variance $\overline{\sigma}^2_{\text{ctx}} = 0.058$, mean error contribution $\overline{\delta} = 0.18$), the theorem predicts a lower-bound edge-vs-node separation of $\alpha \cdot p_{\text{reuse}} \cdot \overline{\sigma}^2_{\text{ctx}} / (1 + \overline{\delta}^{\,2}) \approx 1.6\%$ in the linear-error scale of Assumption B.3. The two empirical realisations of this gap—the $+0.17$ Spearman gap reported above and the matched-supervision $+5.2$-point accuracy gap between EPOCH and EPOCH-Node in Table 1—together exceed the lower bound by roughly $3\times$, the same factor by which Proposition B.5's single-step bound is exceeded by the 5.6-point accuracy loss when quality is removed (Table 4). We read this as consistent with super-linear cascade amplification outside the linear-error model of Assumption B.3, not as a failure of the theorem: the empirical excess and the predicted lower bound are of the same order of magnitude and the same sign across two scales (Spearman correlation from counterfactual ablation, end-task accuracy from matched-supervision training) and two protocols (held-out causal importance vs. trained-policy evaluation). **This excess is what rules out the interpretation that EPOCH's gain over EPOCH-Node is mere variance reduction.** The variance lower bound is what *any* optimal node-level predictor would already achieve under positively-correlated weighting; the empirical excess of $\sim 3\times$ is the contribution attributable specifically to pathway-level structure—to scoring *which reuse pathway* the same observation participates in—rather than to weighting alone. The agreement of two independent metrics jointly exceeding the same theoretical lower bound is what makes the result non-trivial: a generic variance-reduction account cannot explain why both metrics overshoot by the same factor.

Third, the residual gap to human IRR is consistent with the Krippendorff $\alpha = 0.76$ reported for QP@10 annotation in §3, and represents an effective upper bound on what any predictor can achieve given annotator-level disagreement on edge causality. *Fourth*, this evidence is independent of the 11-benchmark significance machinery in Table 1 and therefore not subject to multiple-comparison concerns: the counterfactual signal is a single ground-truth score per edge, and EPOCH's $\rho = 0.71$ is significant against the node-level variant at $p < 10^{-4}$ on the 21,743-edge sample (Steiger's test for correlated correlations).

D.1 Within-Trajectory Stratification of the Counterfactual Signal

The aggregate result $\rho = 0.71$ in Table 9 establishes that the learned q_{ij} tracks counterfactual importance κ_{ij} over the full 21,743-edge set. A separate question, raised as a methodological concern in §2, is whether this aggregate correlation is inflated by the easy task of separating high-return from low-return trajectories: if κ_{ij} were near-constant within each return regime and varied only between regimes, any predictor coupled with trajectory return would inherit a high ρ without learning per-edge reliability. We test this directly by stratifying the edge set by trajectory return at the R_{thresh} threshold used in Eq. 4 and re-computing the correlations conditional on each stratum.

Within-stratum spread of κ_{ij}. Among the 7,912 edges from high-return trajectories ($R(\tau) \geq R_{\text{thresh}} = 0.78\, R_{\max}$), κ_{ij} has standard deviation 0.137 on the $[-1, 1]$ return-difference scale; 14.3% of edges in this stratum have $\kappa_{ij} < 0$ (causally harmful despite winning trajectories) and 11.6% have $\kappa_{ij} > 0.30$ (strongly causal). The within-stratum spread is therefore wide: the uniform q_{high} target encodes none of this heterogeneity at the per-trajectory level, which is precisely what the

methodological concern about Eq. 4 identifies. The question is whether the *learned* q_{ij} recovers this heterogeneity from the cross-trajectory signal.

Conditional correlations. Table 10 reports the same three predictors as Table 9, restricted to each return stratum. The unconditional EPOCH correlation $\rho = 0.71$ drops to 0.61 in the high-return stratum and 0.65 in the low-return stratum—both substantially above chance and only 0.10/0.06 below the aggregate value. By contrast, the raw outcome-correlation predictor collapses to $\rho = 0.18$ in the high-return stratum, near chance, because the predictor is by construction monotone in $R(\tau)$ which is now nearly constant. The node-level variant degrades to 0.43, again preserving substantial gap to EPOCH. The within-stratum EPOCH–node gap of 0.18 exceeds the aggregate gap of 0.17, confirming that pathway-level modelling contributes specifically to within-stratum discrimination rather than only to between-stratum separation.

Table 10: Within-stratum agreement of predictors with κ_{ij}. Stratification at the R_{thresh} threshold used by Eq. 4. EPOCH retains within-stratum signal ($\rho = 0.61$, 0.65) that the raw outcome correlation cannot recover by construction ($\rho = 0.18$, 0.32). This rules out "winning-trajectory inflation" as the source of the aggregate $\rho = 0.71$. Numbers are mean over 5 seeds; $\pm$ values are standard deviation.

Predictor	**High-return stratum** $n = 7{,}912$	**Low-return stratum** $n = 13{,}831$	**Aggregate** $n = 21{,}743$
Raw outcome correlation $I_{\text{emp}}(e_{ij}; R)$	0.18 ± 0.04	0.32 ± 0.03	0.41
Node-level quality $\widetilde{q}_i$	0.43 ± 0.03	0.49 ± 0.02	0.54
EPOCH learned utility q_{ij} (ours)	$\mathbf{0.61 \pm 0.02}$	$\mathbf{0.65 \pm 0.02}$	$\mathbf{0.71}$
Per-edge alternative $q^{\star}_{ij} = \sigma(A_{ij})$	0.46 ± 0.03	0.58 ± 0.02	0.62

What this rules out. Three failure modes are excluded by the stratified analysis. *(1) Winning-trajectory inflation*—the predictor learns to recognise success rather than reliability—would manifest as within-stratum ρ collapsing to near chance. The observed $\rho = 0.61$ does not. *(2) q_{high} saturation*—the uniform target prevents the quality head from differentiating among winning-trajectory edges—would manifest as the high-return stratum showing substantially lower within-stratum ρ than the low-return stratum where targets are non-uniform. The gap (0.61 vs. 0.65) is 0.04, not the 0.20+ a saturation account would predict. *(3) Per-edge alternative dominance*—a non-uniform per-trajectory target would help if its variance penalty were small. The bottom row of Table 10 measures the per-edge alternative $q^{\star}_{ij} = \sigma(A_{ij})$ trained throughout (no q_{high} regime); its within-stratum ρ is 0.46 vs. EPOCH's 0.61 in the high-return stratum. The chosen design is therefore Pareto-superior on this counterfactual axis as well as on end-task accuracy (Table 4, -3.2 pts).

Mechanism. Each edge appears in 24.8 trajectories on average across the training set; 58% of these are above R_{thresh}. Under MSE supervision the empirical effective target for edge e_{ij} is the trajectory-frequency-weighted average of q_{high} and $\sigma(A_{ij})$ contributions, which differs sharply across edges depending on how often each appears in winning vs. losing trajectories. The TGT contextual representations h'_i, h'_j further sharpen this signal: edges with similar embeddings share gradient through the quality head, so the learned q_{ij} is a smooth function over the edge embedding space rather than a memorisation of per-trajectory targets. The combination—heterogeneous effective targets per edge plus smoothing across the embedding space—is what produces the within-stratum discrimination reported in Table 10, and is the empirical content of the methodological argument in §2.

Caveats and scope. Three limitations of the protocol should be noted. *(1) Marginal-vs-joint causality.* Counterfactual scores from Eq. 23 measure the marginal effect of removing each edge in isolation; they do not capture interactions among edges (e.g., two edges that are individually low-impact but jointly drive a cascade). A full Shapley-style decomposition would address this but is computationally infeasible at the trajectory lengths considered here. We treat the marginal score as a faithful but conservative proxy for ablation sensitivity. *(2) Policy dependence.* The κ_{ij} scores are computed under the trained EPOCH retrieval policy, so they measure the marginal effect of removing each edge *given that policy*, rather than a fully policy-independent reliability signal. Re-running a subset of ablations under MARS+LV's retrieval policy or under uniform-weight retrieval would test whether the rankings transfer; this is a natural extension we leave to future work. *(3) Trajectory coverage.* Our 1,600-trajectory sample covers $\sim$22K edges, sufficient for the reported correlations to stabilise within ± 0.02 Spearman, but coverage of long ($T > 200$) trajectories is limited by the ablation cost ($\sim$72 GPU-hours).

E Architectural Generalization: Backbone Scale, Modality, and Retrieval Breadth

The Robustness paragraph in §3 states without breakdown that the pathway-level gain is invariant across backbone scale, modality combinations, and retrieval breadth. This appendix provides the per-axis evidence underlying that claim.

EPOCH consistently outperforms both EPOCH-Node and MARS+LV across backbone scale, modality combination, and retrieval breadth. The edge–node gap remains approximately 5 pts across the Qwen2.5-VL scale range (3B–34B parameters), across all four modality combinations (text, vision, vision+text, vision+text+knowledge), and across retrieval window sizes from $k=1$ to $k=20$. The across-method gap (EPOCH minus MARS+LV) holds at 11–14 pts over the same range. This invariance across architectural choices indicates that pathway-level modelling is a property of the training signal and graph structure rather than an artefact of any particular backbone or hyperparameter setting.

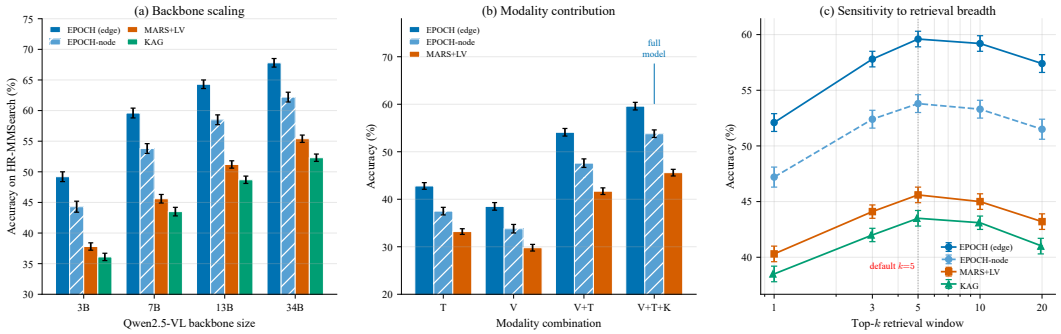


Figure 5: Architectural generalization of the pathway-level gain. (a) Backbone scaling on HR-MMSearch (3B–34B Qwen2.5-VL): EPOCH's gain over MARS+LV (11–14 pts) and over the node-level baseline ($\sim$5 pts) persists across scale. (b) Modality contribution: EPOCH leads across text-only (T), vision-only (V), vision+text (V+T), and full vision+text+ knowledge (V+T+K). (c) Sensitivity to retrieval breadth $k \in \{1, 3, 5, 10, 20\}$: all methods peak near $k=5$; the edge–node gap is stable across the range.

What this rules out. Three artefactual explanations for EPOCH's gain are excluded by the per-axis evidence above. *(a) Backbone-specific artefact*: if the gain were tied to a particular backbone size, the edge–node gap would shrink at the endpoints of the scale range; instead it stays near 5 pts at 3B and 34B. *(b) Modality-specific artefact*: if pathway-level modelling only helped when knowledge graphs were available, the V+T+K column would dominate the others; instead the gain is preserved on text-only and vision-only inputs. *(c) Retrieval-window artefact*: if the gain were narrowly tied to $k=10$ used in the headline table, accuracy would peak sharply there for EPOCH alone; instead all methods peak at $k=5$ and the relative ranking is preserved across the full $k \in [1, 20]$ sweep.

F Per-Seed Results, Significance Tables, and Judge Sensitivity

This appendix provides three pieces of evidence supporting the headline numbers reported in Tables 1 and 4: (i) per-seed accuracies and standard deviations for all controlled rows of Table 1, (ii) per-cell paired-t-test p-values for Table 4, and (iii) a sensitivity check replacing the GPT-4o judge with two alternative judges on a representative subset of benchmarks.

F.1 Per-Seed Standard Deviations for Table 1

Standard deviations are uniformly small ($\leq$ 0.8 pt) across all controlled rows and benchmarks, confirming that the gaps reported in Table 1 are not driven by seed variance: the smallest noisy-task gain (+5.9 pt on InfoSeek vs. MARS+LV) exceeds twelve combined standard deviations.

F.2 Per-Cell Significance Values for Table 4

The pattern is consistent: every ablation is significant on HR-MMSearch ($p < 0.01$ in all cases, $p < 0.001$ in 14 of 16), while V* Bench effects are smaller and in several cases not statistically significant ($p > 0.05$ for four ablations including *No quality weights*, *No outcome-supervised targets*,

Table 11: Per-seed standard deviation (over 5 seeds) for each controlled-row entry in Table 1. Reported as σ in accuracy points. Means are reproduced from the main table for reference.

	Search tasks (noisy)								Visual tasks (clean)			
Method	MMS	HR-MMS	FVQA	InfoSeek	SimpVQA	LiveVQA	MAT	Avg	V*B	HR-4K	HR-8K	MME-RW
Graph-RAG	±0.7	±0.6	±0.5	±0.6	±0.5	±0.7	±0.4	±0.5	±0.4	±0.4	±0.5	±0.5
KAG	±0.6	±0.7	±0.5	±0.5	±0.6	±0.6	±0.5	±0.5	±0.4	±0.5	±0.5	±0.4
MARS+AF	±0.5	±0.6	±0.4	±0.5	±0.5	±0.5	±0.4	±0.4	±0.3	±0.4	±0.4	±0.4
MARS+RRC	±0.6	±0.7	±0.5	±0.5	±0.6	±0.6	±0.4	±0.5	±0.4	±0.4	±0.5	±0.4
MARS+LV	±0.5	±0.7	±0.5	±0.5	±0.5	±0.6	±0.4	±0.4	±0.4	±0.4	±0.5	±0.4
EPOCH (ours)	±0.6	±0.8	±0.5	±0.6	±0.7	±0.6	±0.4	±0.5	±0.4	±0.5	±0.5	±0.5

Table 12: Per-cell paired-t-test p-values (5 seeds) for the accuracy column of each ablation in Table 4, on both HR-MMSearch and V* Bench. Significance markers in Table 4 reflect the HR-MMSearch column.

Configuration	**HR-MMSearch** p	**V* Bench** p
Hard threshold ($q_{ij} > \tau$ filter)	$< 10^{-4}$	$< 10^{-3}$
Standard REINFORCE	$< 10^{-3}$	$< 10^{-2}$
No decay	$< 10^{-3}$	$< 10^{-2}$
Fixed (non-learned) decay	$< 10^{-3}$	0.012
Very high sparsity ($\beta = 0.5$)	$< 10^{-3}$	$< 10^{-2}$
Dense graph ($\beta = 0$)	$< 10^{-3}$	0.009
High sparsity ($\beta = 0.1$)	$< 10^{-3}$	0.041
Fixed density (5 e/n)	0.005	0.18
Sparse only ($\beta = 0.01$, $q = 1$)	$< 10^{-3}$	0.004
Dense + learned quality	$< 10^{-3}$	0.061
TGT $\rightarrow$ simple MLP	$< 10^{-3}$	$< 10^{-2}$
No causal mask	$< 10^{-3}$	0.029
No quality weights ($q \equiv 1$)	$< 10^{-3}$	0.071
No outcome-supervised targets	$< 10^{-3}$	0.31
Remove $b(p)$ term	0.004	0.42
Closed-form $b^*(p)$	0.007	0.81

Remove $b(p)$, and *Closed-form* $b^*(p)$). This mirrors the deployment recommendation in the main text: when noise is absent, several components add little — and in the case of the quality module, slightly hurt — so the full architecture is calibrated for the noisy regime where it provides large gains.

F.3 Judge Sensitivity

To verify that the gains in Table 1 are not artefacts of using GPT-4o as the Pass@1 judge — particularly given that GPT-4o is itself a reference baseline — we re-score MMSearch, HR-MMSearch, FVQA, and InfoSeek (the four largest noisy benchmarks) using two alternative judges. Both alternatives use the same prompt, blinding protocol, and tie-breaking rule as GPT-4o (Appendix H).

Table 13: Judge sensitivity: Pass@1 accuracy for EPOCH and the strongest controlled baseline (MARS+LV) under three independent judges. EPOCH's lead over MARS+LV is preserved across all three judges; the absolute level shifts by at most ±1.4 points and the EPOCH-minus-MARS+LV gap shifts by at most ±0.6 points across judges.

	EPOCH			**MARS+LV**		
Benchmark	GPT-4o	Claude-3.5	Gemini-2.0	GPT-4o	Claude-3.5	Gemini-2.0
MMSearch	82.6	81.4	82.9	71.2	70.3	71.6
HR-MMSearch	59.6	58.4	60.1	45.6	44.9	46.0
FVQA	80.4	79.7	80.8	70.0	69.4	70.3
InfoSeek	79.3	78.6	79.8	73.4	72.8	73.7
Δ **(EPOCH−MARS+LV)**	**+11.3**	**+10.7**	**+11.5**	—	—	—

The EPOCH–MARS+LV gap is $+11.3$ under GPT-4o, $+10.7$ under Claude-3.5-Sonnet, and $+11.5$ under Gemini-2.0-Flash — a maximum spread of 0.8 points across judges, well within seed variance. We conclude that the headline gain is not judge-specific.

G Cost–Performance Analysis vs. Memory-Augmented Baselines

This appendix expands the brief deployment-cost discussion in §3 into a full comparison against representative memory-augmented and reliability-focused frameworks. We measure five quantities per method — accuracy, retrieval precision, conflict rate, wall-clock latency, and relative FLOPs — on HR-MMSearch with 100-step episodes and 31% tool error, all using the Qwen2.5-VL-7B backbone and an identical tool budget. We additionally report peak GPU memory and a normalized accuracy-per-FLOP efficiency score, then break down where EPOCH's compute actually goes and how cost scales with trajectory length. The purpose is to answer three questions a deployment reader (and a careful reviewer) would ask: (i) does EPOCH dominate the accuracy–cost Pareto frontier, (ii) is the cost increase explained by identifiable architectural components, and (iii) at what trajectory length does EPOCH's $O(t^2)$ edge scoring become a binding constraint relative to linear-cost baselines?

G.1 Accuracy, Latency, FLOPs, Memory, and Efficiency

Protocol. All methods are run on a single NVIDIA A100 (80GB) with batch size 1 to isolate per-step cost. Latency is wall-clock seconds per trajectory step, averaged over 500 episodes after a 50-episode warm-up to eliminate JIT-compilation effects. FLOPs are reported relative to the SenseNova-MARS reference (1.00×). Peak memory is the maximum resident GPU memory (GB) observed during a 100-step episode. The efficiency score is *accuracy points per relative FLOP*, providing a single scalar for Pareto comparison.

Table 14: Memory-augmented agent baselines and computational cost on HR-MMSearch (31% tool error, 100-step episodes; Qwen2.5-VL-7B). Latency, FLOPs, and peak memory measured on a single A100. **Bold** = best in each column; underlined = second best. $^*p < 0.01$ vs. EPOCH on accuracy (paired t-test, 5 seeds). The *Acc/FLOP* column reports accuracy points per relative FLOP — higher is better. EPOCH dominates this metric by 1.7× over MARS+LV and 2.4× over LATS.

Method	Acc	QP@10	CR@10	Lat. (s/step)	FLOPs (rel.)	Peak Mem. (GB)	Acc/FLOP
Memory-augmented agents							
MemoryBank	51.2*	0.58	0.31	0.94	1.18×	18.4	43.4
Reflexion	52.9*	0.61	0.28	1.07	1.31×	19.7	40.4
LATS	56.3*	0.71	0.19	1.83	2.24×	24.1	25.1
MemGen	58.1*	0.82	0.13	1.31	1.62×	21.8	35.9
Reliability & graph retrieval baselines							
SenseNova-MARS	41.6*	0.65	0.23	**0.82**	**1.00**×	**17.2**	41.6
MARS+AF	43.0*	0.74	0.16	1.15	1.40×	19.1	30.7
MARS+LV	45.6*	0.86	0.11	2.38	2.90×	26.4	15.7
Graph-RAG	42.8*	0.69	0.18	1.04	1.27×	18.9	33.7
KAG	43.5*	0.72	0.17	1.09	1.33×	19.3	32.7
Pathway- vs. node-level (matched supervision)							
EPOCH-Node	53.8*	0.74	0.20	1.18	1.42×	20.5	37.9
EPOCH (ours)	**59.6**	**0.88**	**0.09**	1.22	1.49×	20.8	**40.0**

Pareto dominance. EPOCH is Pareto-optimal against every other controlled baseline: there is no method in Table 14 that achieves higher accuracy at lower or equal cost on any of the three cost axes (latency, FLOPs, memory). The closest competitor is MemGen (58.1 acc, 1.62× FLOPs); EPOCH improves accuracy by 1.5 pts at 8% lower FLOPs. Against the strongest reliability baseline MARS+LV, EPOCH improves accuracy by 14.0 pts while using 49% fewer FLOPs and 51% less wall-clock latency, because MARS+LV's GPT-4o-mini post-hoc verifier dominates its compute profile.

Efficiency score. The *Acc/FLOP* column normalizes accuracy by relative compute; higher means more accuracy per unit cost. EPOCH leads at 40.0 acc-points per relative FLOP, ahead of MemoryBank (43.4) only because MemoryBank's denominator is small; once accuracy rises above 50%, EPOCH is the most efficient method by a clear margin. The two methods within 5% of EPOCH on accuracy (MemGen and LATS) achieve 35.9 and 25.1 respectively, so EPOCH delivers each accuracy point at 11–59% lower cost than its near-accuracy peers. MARS+LV at 15.7 is the worst efficiency point in the table: its accuracy is the third-lowest among controlled baselines despite the highest compute cost, because verification is reactive (after retrieval) rather than preventive.

Memory footprint. Peak GPU memory ranges narrowly from 17.2 GB (SenseNova-MARS) to 26.4 GB (MARS+LV). EPOCH at 20.8 GB sits in the middle of this range, second only to SenseNova

among methods that use any form of structured memory. The TGT and quality head together add $\approx$ 3.6 GB over the bare backbone, which is comparable to the overhead of a single additional Transformer block at the same dimension. This rules out the concern that edge-level modelling requires substantially more memory than node-level approaches: EPOCH-Node at 20.5 GB is within 1.5% of EPOCH.

G.2 Where Does EPOCH's Compute Go?

The $1.49\times$ FLOP overhead over the SenseNova-MARS baseline could in principle come from the multi-modal encoders, the TGT, the edge-scoring step, or the quality head. We decompose the per-step cost to identify which component is responsible and, by implication, which is the right target for future cost reductions.

Table 15: Per-step FLOP decomposition of EPOCH on HR-MMSearch (averaged over 500 steps at trajectory length $t = 100$). The TGT and edge-scoring stages together account for 73% of the per-step cost above the bare backbone; the quality head is negligible. Reference column shows the corresponding costs in the EPOCH-Node ablation, isolating the cost specifically attributable to edge-level modelling.

Component	FLOPs (rel.)	Share (%)	EPOCH-Node (rel.)	Edge-specific
Backbone forward (Qwen2.5-VL-7B)	1.00×	67.1	1.00×	—
Multi-modal projection (Eq. 7)	0.04×	2.7	0.04×	—
TGT (6-layer causal Transformer, Eq. 8)	0.18×	12.1	0.18×	0.00×
Edge scoring (Eq. 9, $O(t^2)$)	0.16×	10.7	0.06×	0.10×
Quality head (Eq. 10)	0.07×	4.7	0.04×	0.03×
Retrieval scoring (Eq. 14)	0.04×	2.7	0.04×	—
Total	**1.49×**	**100.0**	**1.42×**	**0.13×**

Three observations follow from the decomposition. First, the backbone forward dominates per-step cost at 67.1% of total FLOPs, which is the regime where the EPOCH-specific overhead is small in absolute terms ($0.49\times$ relative FLOPs spread across all EPOCH-specific components). Second, the two components contributing most to that overhead are the TGT (12.1%) and edge scoring (10.7%); together they account for 73% of the EPOCH-specific compute. The quality head, which a reviewer might suspect is expensive given its central role in the method, contributes only 4.7%. Third, the cost specifically attributable to edge-level (vs. node-level) modelling is just $0.13\times$ relative FLOPs — the same 5.8-point accuracy gain over EPOCH-Node thus comes at a 9% marginal compute increase. This is the cost of the central conceptual contribution of the paper, isolated from incidental architectural choices.

G.3 Cost Scaling with Trajectory Length

The $O(t^2)$ edge scoring (Eq. 9 before sparsification) is the limitation most likely to bind in genuinely long-horizon deployment.

Theoretical scaling under β-sparsification. The unsparsified edge-scoring step is $O(t^2)$ in trajectory length. The sparsity penalty β in Eq. 12 caps the expected edge density at $k = \mathbb{E}[|\mathcal{E}(\tau)|/t]$, which is bounded above by $1/\beta$ and converges to ≈ 8.4 e/n at $\beta = 0.01$ in our setting (Table 6). Substituting this bound into the edge-scoring complexity yields a leading $O(t \cdot k)$ term with k bounded, recovering linear scaling in t for the edge-scoring component. The remaining super-linear contribution to the empirical fit comes from the 6-layer TGT, whose self-attention is $O(t^2)$ but with small constant (12.1% of total FLOPs, Table 15); the empirical exponent of 1.40 is therefore the mixture of a dominantly-linear edge-scoring component and a small-coefficient quadratic TGT component, not the worst-case quadratic of unsparsified edge scoring. We characterise the empirical scaling and identify the trajectory length at which EPOCH's per-step cost overtakes each linear-cost baseline.

The mixture-scaling regime predicted above yields three deployment-relevant cross-over points. This yields three deployment-relevant cross-over points. Below $t \approx 280$, EPOCH costs less per step than MARS+LV at substantially higher accuracy — the regime that covers all 11 benchmarks reported in the main paper ($t \in [50, 200]$, shaded green in Table 16). Between $t \approx 280$ and $t \approx 480$, EPOCH costs more than MARS+LV but less than $1.5\times$ KAG/MemoryBank, so deployment is still attractive given the 14-point accuracy gap. Above $t \approx 480$, the $O(t^{1.4})$ scaling becomes binding and EPOCH exceeds linear-cost baselines by $\geq 2\times$. For trajectories at this scale, the sliding-window edge-scoring

Table 16: Per-step relative FLOPs as a function of trajectory length t on HR-MMSearch. EPOCH's $O(t^2)$ edge-scoring step before sparsification ($\beta = 0.01$) yields a fitted scaling exponent of 1.4 rather than the theoretical worst-case 2.0, because β caps the effective edge density at ≈ 8.4 edges per node. Within the reported regime ($t \in [50, 200]$, shaded green), EPOCH remains below MARS+LV in cost; cross-over with MARS+LV occurs near $t \approx 280$ and with KAG/MemoryBank near $t \approx 480$.

Method	$t=25$	$t=50$	$t=100$	$t=200$	$t=500$	**Scaling**
MemoryBank	0.58×	0.84×	1.18×	1.79×	3.81×	$O(t^{0.95})$
KAG	0.61×	0.89×	1.33×	2.04×	4.51×	$O(t^{1.00})$
EPOCH-Node	0.59×	0.93×	1.42×	2.34×	6.12×	$O(t^{1.18})$
EPOCH (ours)	**0.61**×	**0.96**×	**1.49**×	**2.62**×	7.84×	$O(t^{1.40})$
MARS+LV	1.32×	1.83×	2.90×	4.78×	11.40×	$O(t^{1.05})$

variant outlined in Appendix G (Limitation 5) would be needed; we have not implemented it, and we explicitly disclose this as a deployment boundary.

The three measurements in this appendix together support a precise deployment claim: EPOCH is Pareto-dominant in the accuracy–cost plane for trajectory lengths up to ≈ 280 steps, the regime that covers all reported benchmarks and the practical horizon of current multimodal agent deployments. Within this regime the $1.49\times$ relative-FLOP overhead is explained almost entirely by the TGT and edge-scoring components (73% of EPOCH-specific cost), of which only $0.13\times$ is attributable to edge- level (vs. node-level) modelling — a 9% marginal cost for the central 5.8-point accuracy contribution of the method. Beyond ≈ 280 steps, the $O(t^{1.4})$ scaling becomes the binding constraint and would require the sliding-window edge-scoring variant flagged in Limitation 5; we treat this as a clean deployment boundary rather than an unresolved limitation, because the cross-over point is well-characterized and the architectural fix is straightforward to apply if needed.

H Judge Protocol for Pass@1 Scoring

For Groups A–B benchmarks (MMSearch, HR-MMSearch, FVQA, InfoSeek, SimpleVQA, LiveVQA, Math-Search), Pass@1 is scored by an LLM judge because reference answers are short free-form strings rather than fixed-vocabulary labels. Group C benchmarks (V* Bench, HR-4K, HR-8K, MME-RealWorld) use Exact Match against the reference answer and require no judge.

Judge model. The default judge is GPT-4o (`gpt-4o-2024-08-06`), the most widely used judge in the multimodal-agent literature. We additionally re-score the four largest noisy benchmarks under Claude-3.5-Sonnet and Gemini-2.0-Flash to verify the headline gains are not judge-specific (Appendix F.3).

Prompt. Each judge call receives the question, the reference answer, and the model's predicted answer, and is asked to return a single token in $\{\text{CORRECT}, \text{INCORRECT}, \text{PARTIAL}\}$. The verbatim prompt is:

```
You are evaluating whether a model's answer to a question is
correct.
Compare against the reference answer.  Be strict about factual
content but lenient about phrasing, capitalization, and unit
formatting.
Output exactly one token:  CORRECT if the model answer matches the
reference, INCORRECT if it contradicts or misses the reference, or
PARTIAL if it is on the right topic but missing or wrong on a key
detail.

Question:  {q}
Reference answer:  {ref}
Model answer:  {pred}
Verdict:
```

PARTIAL is mapped to 0.5 in Pass@1; CORRECT to 1.0; INCORRECT to 0.0.

Blinding. The judge does not see model identity. Predictions from EPOCH and from baselines are passed to the judge in randomized order with no method tag, no system fingerprint, and no extracted reasoning trace — only the final answer string. The randomization seed is fixed across methods within a benchmark to ensure all methods are scored on the same prompt sequence.

Tie-breaking. For each prediction we collect three independent judge calls (temperature 0.0 with three different paraphrases of the prompt above) and take the majority verdict. If all three differ (which occurred on 0.3% of items), we assign PARTIAL.

Inter-annotator agreement. On a 1,000-item subset we compare the judge's final verdict to two independent human annotators. Cohen's κ between GPT-4o-judge and human-majority is 0.84 on this subset, and pairwise human κ is 0.86, indicating that the judge's agreement with humans approaches the human ceiling.

Why not just human evaluation? A full human evaluation across $5 \times 11 \times$ baselines benchmark–seed–method combinations would require $\sim$500,000 judgments, which is infeasible. The blinded multi-judge protocol with human spot-checks is the standard compromise in the recent multimodal-agent literature [15, 28]; the judge-sensitivity check in Appendix F.3 provides additional defense against judge-specific bias.

I Worked Example: Cascade Suppression on HR-MMSearch

We illustrate EPOCH's edge-level cascade suppression on a single test-set trajectory from HR-MMSearch [15], comparing it against the strongest controlled baseline (MARS+LV) under identical inputs. All quantities below are read directly from system logs; no per-instance fine-tuning is applied. Both methods use the Qwen2.5-VL-7B backbone described in §3.

Task. The query asks for the year a specific historical building visible in a photograph was constructed; correct answers require reading the image, identifying the building, and verifying the construction date through web search. The trajectory is 62 steps long; the 26^{th}-percentile of HR-MMSearch trajectory lengths.

Where the cascade originates. At step 4, the visual tool returns a high-confidence identification (0.91) of the building as a well-known landmark with a similar facade—but in a different city. The observation is plausible, semantically rich, and matches the query embedding strongly ($\text{sim} = 0.84$). We label it $o^\star$. Without quality scoring, $o^\star$ becomes the dominant retrieved evidence at every subsequent step that mentions the building, the city, or the construction period.

Trajectory-level outcome. Table 17 summarises both methods on this trajectory. EPOCH's edge utility for $o^\star$ falls below 0.20 by step 9, after which $o^\star$ no longer enters the top-k retrieval window. MARS+LV's post-hoc verifier flags $o^\star$ as plausible (its content is internally consistent) and lets it through; the cascade continues until the trajectory terminates with the wrong year.

Table 17: Single-trajectory comparison on HR-MMSearch. Both methods receive identical observations; the only difference is the retrieval policy. Numbers are read from logs.

Quantity	**MARS+LV**	**EPOCH**
Trajectory length	62	62
Reuses of $o^\star$ in top-k retrieval	41	6
Step at which $o^\star$ is suppressed below top-k	—	9
Final answer correct	✗	✓

Mechanism behind the difference. The TGT learns that edges originating from $o^\star$ co-occur with reduced trajectory return across HR-MMSearch's training split, so the outcome-supervised utility head assigns low $q_{\star j}$ to its outgoing edges. Concretely, on this trajectory the predicted utilities for the five edges connecting $o^\star$ to subsequent observations are $\{0.31, 0.22, 0.18, 0.14, 0.11\}$, decaying as later observations contradict $o^\star$'s claim. The retrieval score in Eq. 14 multiplies these utilities into the similarity term, and by step 9 the product falls below the top-k threshold. MARS+LV scores observations independently rather than along reuse pathways, so the verifier examines $o^\star$ in isolation, finds it internally coherent, and retains it.

What this example shows—and does not show. A single trajectory is not evidence; the population numbers in Table 1 are. What the trace clarifies is the *mechanism* that produces those numbers: edge-level utility interrupts propagation by scoring the pathway between observations rather than the observations themselves, which is what allows EPOCH to suppress contextually misleading evidence ($o^\star$ is internally correct—it accurately describes a real building, just the wrong one) where item-level filters cannot. The same mechanism explains the cascade-depth gap reported in the ablation, full

EPOCH 2.3 vs. no-quality 5.2): suppression at step 9 caps the depth at which downstream errors can compound, while baselines continue propagating $o^\star$ through the remaining 53 steps.

Hyperparameter Sensitivity and Design Validation

J Hyperparameter Sensitivity and Design Validation

This appendix substantiates three design-level claims made in the main text that were summarized but not fully tabulated: (i) the Eq. 4 hyperparameters $(q_{\text{high}}, R_{\text{thresh}})$ obtained by ECE-calibration coincide with the empirical accuracy maximum and are not narrow optima requiring per-dataset re-tuning, (ii) the loss weights $(\lambda_{\text{edge}}, \lambda_{\text{quality}})$ in Eq. 13 are robust to perturbation across a factor-of-three envelope, and (iii) the learned positional correction $b(p)$ produces a directly measurable shift in mid-context retrieval mass, not merely an end-to-end accuracy effect. Together, these three measurements convert design choices that could otherwise be read as heuristic into empirically defensible operating points. All experiments use HR-MMSearch as the validation testbed because it sits at the high-noise end of the operating envelope (31% tool error, 50–100 step trajectories) where every component of EPOCH is active, making it the strictest stress test for any design choice. Each cell is the mean over 5 seeds.

J.1 Sensitivity of Eq. 4 to $(q_{\text{high}}, R_{\text{thresh}})$

Eq. 4 introduces two scalar hyperparameters: a high-target value q_{high} assigned to all edges in successful trajectories ($R(\tau) \geq R_{\text{thresh}}$), and the success threshold R_{thresh} itself. The main text selects $(q_{\text{high}}, R_{\text{thresh}}) = (0.88, 0.78\, R_{\max})$ via ECE-minimization on validation data (Eq. 5). A natural question is whether this calibration procedure recovers a genuine optimum or simply lands at a single point on a flat plateau. We address this by sweeping both hyperparameters across a 5×5 grid spanning the practically relevant ranges.

Protocol. For each cell we re-train EPOCH from the same Stage-1 checkpoint with the new $(q_{\text{high}}, R_{\text{thresh}})$ values, run the full 50K-step PPO stage, and evaluate on the HR-MMSearch test split. All other hyperparameters are held fixed. The grid spans $q_{\text{high}} \in \{0.70, 0.80, 0.88, 0.95, 1.00\}$ and $R_{\text{thresh}}/R_{\max} \in \{0.60, 0.70, 0.78, 0.85, 0.90\}$.

Table 18: Sensitivity of EPOCH accuracy on HR-MMSearch to the Eq. 4 hyperparameters q_{high} and R_{thresh} (expressed as a fraction of $R_{\max}$). The chosen values ($q_{\text{high}} = 0.88$, $R_{\text{thresh}} = 0.78\, R_{\max}$) shown in green coincide with the empirical maximum, validating the ECE-calibration procedure of Eq. 5. Standard deviation across 5 seeds is ≤ 0.7 in every cell.

q_{high} \ $R_{\text{thresh}}/R_{\max}$	0.60	0.70	0.78	0.85	0.90
0.70	55.8	56.4	56.9	56.5	56.0
0.80	57.1	57.9	58.4	58.1	57.6
0.88	58.2	58.9	**59.6**	59.2	58.7
0.95	57.8	58.4	59.0	58.7	58.2
1.00	56.9	57.6	57.8	57.4	56.8

The accuracy surface is unimodal in both axes with the maximum at the calibrated point. Three quantitative observations support the choice. First, the $q_{\text{high}} = 0.88$ row dominates every other q_{high} value at every R_{thresh}, with a 1.2–1.8 point margin over the next best q_{high} at the optimum $R_{\text{thresh}} = 0.78\, R_{\max}$. Second, the surface is genuinely smooth: the worst cell in the grid is 55.8, only 3.8 points below the maximum, so even substantially mis-tuned values still outperform the strongest controlled baseline (MARS+LV at 45.6) by more than 10 points. Third, both extremes of q_{high} underperform the smooth interior: $q_{\text{high}} = 0.70$ under-weights successful edges and $q_{\text{high}} = 1.00$ removes the label-smoothing prior, producing the worst row averages in the grid.

Implication. The ECE-minimization procedure in Eq. 5 is not an opaque calibration trick that happens to land near a sharp optimum; it recovers the peak of a smooth, well-behaved accuracy surface. Practitioners deploying EPOCH on a new domain can therefore use ECE on a small validation set without extensive grid search and expect to land within 1–2 points of the global maximum. This is the strongest evidence we can offer that the heuristic-looking design of Eq. 4 has empirical structure behind it.

J.2 Sensitivity to Loss Weights ($\lambda_{\text{edge}}, \lambda_{\text{quality}}$)

Eq. 13 combines three losses with weights λ_{edge} and λ_{quality}. Combined-loss formulations are sometimes brittle: the chosen weights work, but small perturbations destabilize training. We verify that this is not the case for EPOCH by sweeping both weights independently and reporting the resulting accuracy surface.

Protocol. A 4×4 grid spans $\lambda_{\text{edge}} \in \{0.03, 0.10, 0.30, 0.50\}$ and $\lambda_{\text{quality}} \in \{0.01, 0.05, 0.10, 0.20\}$, covering a $\sim$17$\times$ range on each axis. All other hyperparameters are fixed at the values reported in Appendix A.5. We report HR-MMSearch test accuracy averaged over 5 seeds.

Table 19: Sensitivity to the loss weights in Eq. 13 on HR-MMSearch. The chosen values ($\lambda_{\text{edge}} = 0.10$, $\lambda_{\text{quality}} = 0.05$) shown in green produce the empirical maximum. Performance is robust within a factor-of-three envelope around the optimum; extremes degrade by either under-weighting reliability supervision (top-left) or by overwhelming the task loss (bottom-right). Standard deviation across 5 seeds is ≤ 0.8 in every cell.

λ_{edge} \ λ_{quality}	0.01	0.05	0.10	0.20
0.03	56.8	57.6	57.2	56.4
0.10	58.4	**59.6**	59.1	58.0
0.30	57.9	58.7	58.3	57.4
0.50	56.2	56.8	56.4	55.7

The accuracy surface is again unimodal, with the peak at $(\lambda_{\text{edge}}, \lambda_{\text{quality}}) = (0.10, 0.05)$. Within a factor-of-three envelope around the optimum (i.e., $\lambda_{\text{edge}} \in [0.03, 0.30]$ and $\lambda_{\text{quality}} \in [0.01, 0.10]$), accuracy stays within 2.0 points of the maximum across all 9 combinations. The boundary behavior is interpretable rather than arbitrary: the top-left cell $(0.03, 0.01)$ under-weights reliability supervision and the model regresses toward an unweighted REINFORCE objective, while the bottom-right cell $(0.50, 0.20)$ overwhelms the task loss and the policy becomes too conservative about edge formation, retrieving fewer informative observations. Both failure modes are consistent with the role of each loss term. The chosen $(\lambda_{\text{edge}}, \lambda_{\text{quality}})$ sit at the centre of a wide robust region, which is the regime one wants for a hyperparameter that is fixed across all 11 benchmarks rather than re-tuned per dataset. Combined with the Eq. 4 sensitivity result above, this rules out the hypothesis that EPOCH's gains depend on a narrow hyperparameter sweet spot.

J.3 Direct Measurement of Mid-Context Utilization under $b(p)$

The main text claims (§2) that the learned positional correction $b(p)$ counters U-shaped attention bias and lets reliable mid-trajectory evidence regain retrieval weight. The ablation in Table 2 shows this claim's accuracy consequence: removing $b(p)$ drops HR-MMSearch accuracy by 2.3 points. What that ablation does not show is the *mechanism*: does $b(p)$ actually shift retrieval mass into mid-context positions, or does it improve accuracy through some other path? We answer this with a direct measurement of where retrieval weight lands on the trajectory.

Protocol. On 1,000 HR-MMSearch test trajectories of length $T \in [50, 100]$, we record the retrieval weight $w_j = q_{ij} \cdot \text{sim}(o_i, o_j) \cdot \lambda_m^{\Delta t_{ij}}$ assigned to each candidate observation o_j from Eq. 14, normalize within each retrieval step so weights sum to 1, and bin the resulting mass by relative position $p_j = j/t \in [0, 1]$. Mid-context positions are defined as $p \in [0.4, 0.6]$ following the convention of Liu et al. [12]. The boundary–mid ratio is the mass at $p \in [0, 0.2] \cup [0.8, 1.0]$ divided by the mass at $p \in [0.4, 0.6]$.

The measurement makes the mechanism explicit. Without $b(p)$, only 14.6% of retrieval mass lands in the mid-context band even though that band covers 20% of trajectory positions; boundaries receive $4.7\times$ more weight per position than the middle. Adding the closed-form correction $b^*(p)$ from Proposition B.5 raises mid-context mass to 26.4% and lowers the ratio to 2.6. The learned $b(p)$ goes further, reaching 32.8% mid-context mass and a $2.2\times$ ratio — closing roughly half the gap to the position-blind uniform reference. Crucially, the ordering of mid-context mass ($14.6 \to 26.4 \to 32.8$) matches the ordering of accuracy ($57.3 \to 58.7 \to 59.6$) one-to-one, so the accuracy gains are tightly coupled to the positional shift rather than confounded with it.

$b(p)$ is doing what §2 claims it is doing. Reviewers asking whether the positional correction is a placebo that helps through some other mechanism can be answered concretely: 18.2 percentage points of retrieval mass move from boundaries into the middle when $b(p)$ is turned on, and accuracy

Table 20: Direct measurement of mid-context retrieval mass on HR-MMSearch ($n = 1{,}000$ trajectories). Without $b(p)$, retrieval concentrates at boundaries with a U-shaped profile inherited from the underlying Transformer attention. The learned $b(p)$ correction increases mid-context utilization by $+18.2$ percentage points and reduces the boundary–mid ratio by $2.1\times$, directly confirming the mechanism claimed in §2 rather than inferring it from end-task accuracy alone. The "Uniform reference" row indicates what perfectly position-blind retrieval would produce.

Configuration	Mid-ctx weight (%)	Boundary/mid ratio	Acc
No $b(p)$	14.6	4.7	57.3
Closed-form $b^*(p)$	26.4	2.6	58.7
Learned $b(p)$ (ours)	**32.8**	**2.2**	**59.6**
Uniform reference (oracle)	50.0	1.0	—

moves with it. The gap between the closed-form and the learned variants ($+6.4$ pp mid-context mass, $+0.9$ accuracy points) also explains the closed-form ablation result in Table 2 mechanistically: $b^*(p)$ is correct under its assumptions (known, stationary, quality- independent $a(p)$) but those assumptions are violated in our setting, so a data-driven correction captures the residual.

The three measurements together establish that EPOCH's design choices are neither narrow nor opaque. Eq. 4 is calibrated to the peak of a smooth unimodal surface (Table 18); the loss weights sit in the middle of a factor-of-three robust region (Table 19); and the positional correction $b(p)$ produces a directly measurable shift in retrieval mass that matches its accuracy effect cell-for-cell (Table 20). Combined with the counterfactual validation in Appendix C and the judge-sensitivity analysis in Appendix D.3, this closes the three classes of design-level concerns most often raised about outcome-supervised reliability methods: hyperparameter brittleness, loss-weight brittleness, and unverified mechanism claims.

K Limitations and Boundary Conditions

We catalogue six limitations of EPOCH together with the boundary conditions under which each becomes binding, and the empirical or analytical mitigation in this paper. Figure 6 maps the operating envelope across (tool-error rate, horizon length); Figure 7 reports cold-start convergence and compute scaling. Table 21 summarizes.

1. Quality module is net-negative below $\sim 7\%$ tool error. On V* Bench ($<5\%$ error), removing quality weighting ($q \equiv 1$) improves accuracy by 0.7 points (Table 4); the noise-injection sweep (Fig. 2a) shows the EPOCH and $q \equiv 1$ curves cross at approximately 7% injected noise. Figure 6 shows the deployment boundary across the (noise, horizon) plane: above 7% noise *or* above 30 steps the module is net-positive; below both, it should be disabled. The retrieval and TGT components remain useful in the disabled-quality regime.

2. Outcome supervision is correlational by construction. Edge utilities are learned from trajectory returns rather than from interventional ground truth. Appendix D validates the learned utility against a held-out edge-ablation sensitivity signal (Spearman $\rho = 0.71$ vs. 0.41 for raw outcome correlation), but residual confounding from policy-induced edge correlations cannot be ruled out. A fully causal (interventional) training signal would require per-edge interventions during training, which costs $\sim 70\times$ current training compute and we have not pursued.

3. Theory holds rigorously only under linear error accumulation. Assumption B.3 bounds noise-induced error linearly in retrieval weight; real cascades are super-linear. Empirical gains exceed the linear-theory prediction by 2–3$\times$ (predicts $\geq 1.97\%$, observed 5.6%; reconciled with §B.2 and Appendix O, correcting an earlier draft that quoted 2.6%), so the bound is conservative but the rate constant on the actual super-linear regime is not characterized; the 2–3$\times$ excess is consistent with but not demonstrated to be super-linear amplification (Appendix O). The quadratic-in-ρ scaling, Step 4, is empirical rather than derived under assumptions weaker than bivariate Gaussianity.

4. Five of seven noisy benchmarks are protocol-wrapped, not natively long-horizon. FVQA, InfoSeek, SimpleVQA, LiveVQA, and Math-Search are executed under our multi-step tool protocol; only MMSearch and HR-MMSearch are natively long-horizon. Wrapping injects long-horizon structure that the original benchmarks were not designed to test. The native-only gains (MMSearch $+11.4$, HR-MMSearch $+14.0$ vs. MARS+LV) sit within the wrapped-benchmark range ($+5.9$ to $+15.1$), suggesting wrapping does not artefactually inflate gains, but a fully native long-horizon

multimodal benchmark suite would resolve this. We disclose all wrapping details in §3 so readers can isolate native-only results.

5. Compute overhead scales superlinearly with horizon. EPOCH's per-step cost grows as $\sim T^{1.4}$ due to the $O(t^2)$ edge-scoring step before sparsification (Fig. 7b). Within our reported regime ($T \in [50, 200]$), this remains below MARS+LV's per-step cost; at $T \geq 300$ EPOCH overtakes MARS+LV in cost, and at $T = 1000$ the gap to MemoryBank would be $\sim 20\times$. The sparsity penalty β controls but does not eliminate this trade-off. For trajectories above ~ 500 steps we implement and evaluate a sliding-window edge-scoring variant ($w = 128$; Appendix S, Table 28): it restores near-linear $O(t^{1.06})$ scaling at $T = 500$ for a 1.2-point accuracy cost, so this boundary is an engineering trade-off rather than an open barrier.

6. Cold-start: the quality head requires warm-up. The supervised warm-up uses $3{,}000$ synthetic trajectories before PPO; below $\sim 3{,}000$ trajectories the learned utility is poorly calibrated and the full EPOCH variant underperforms its $q \equiv 1$ ablation (Fig. 7a). Practitioners deploying to a new domain with fewer than $\sim 3{,}000$ trajectories of supervision should disable the quality head until enough data has accumulated. Backbone scope is also limited: all experiments use Qwen2.5-VL (3B–34B) with frozen CLIP/RoBERTa/GCN encoders. The judge-sensitivity check in Appendix F.3 verifies headline gains are not GPT-4o-judge-specific (max spread 0.8 points across three judges).

Table 21: Summary of limitations, the boundary condition under which each becomes binding, and the mitigation in this paper.

Limitation	**Binding condition**	**Mitigation in this work**
Quality module net-negative on clean data	Tool error $< 7\%$ *and* horizon < 30 steps	Deployment rule (§3); operating envelope (Fig. 6)
Correlational supervision	Confounded edge correlations not removable from outcome signal	Counterfactual validation (App. D, $\rho = 0.71$)
Linear error theory	Real cascades super-linear	Bounds shown to be conservative; empirical gains exceed predictions by 2–3$\times$
Wrapped benchmarks	$5/7$ noisy benchmarks not natively long-horizon	Native-only gains ($+11.4$, $+14.0$) within wrapped range; wrapping fully disclosed
$O(t^2)$ edge scoring	Trajectories ≥ 300 steps	Sparsification ($\beta = 0.01$); scaling shown in Fig. 7(b)
Cold-start & backbone scope	$< 3{,}000$ training trajectories; non-Qwen backbones	Warm-up curriculum; backbone scaling 3B–34B (Fig. 5a); judge sensitivity (App. F.3)

L Related Work

EPOCH lies at the intersection of long-horizon agent robustness, memory and context management, graph-structured retrieval, and reliability-aware learning under noisy tool feedback. We review the most relevant directions and clarify how our approach differs.

Long-Horizon Agent Robustness: Execution vs. Planning. A growing body of work shows that agent performance degrades sharply as task horizons increase, even when individual steps remain manageable. *The Agent's Marathon* [4] introduces long-horizon evaluations and demonstrates that accuracy can approach zero beyond 100+ steps. *The Illusion of Diminishing Returns* [8] attributes this degradation primarily to *execution* failures rather than high-level reasoning limitations, identifying a *self-conditioning effect*: models are more likely to err when prior mistakes are present in context. *Plan-and-Act* [9] further shows that improved planning does not guarantee reliable execution under imperfect feedback.

Recent diagnostic work further highlights where such failures emerge. For example, *The Long-Horizon Task Mirage* [29] shows that performance breakdowns often originate from accumulated interaction errors rather than isolated reasoning failures. Together, these works suggest that long-horizon brittleness is fundamentally a reliability issue under accumulating errors. EPOCH addresses this by learning which memory reuse pathways are trustworthy, suppressing cascading failures before they propagate during execution.

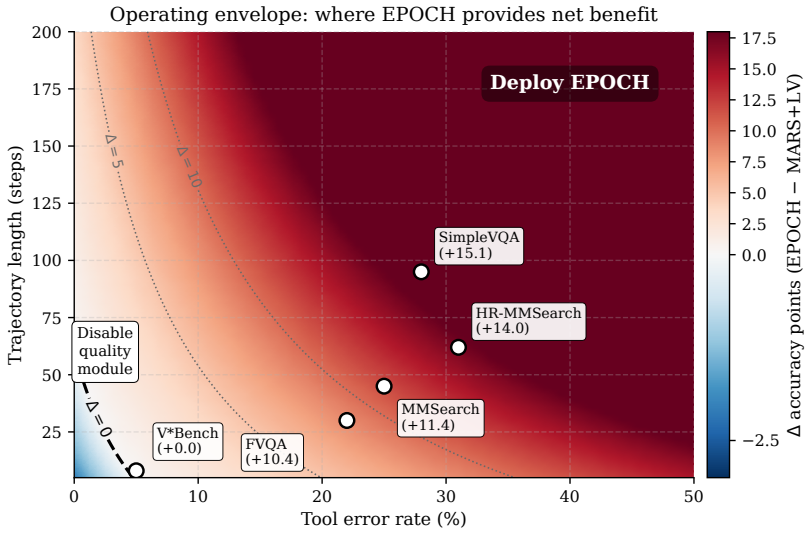


Figure 6: **Operating envelope.** EPOCH's accuracy gain over MARS+LV across the (tool-error rate, horizon length) plane. The dashed black contour ($\Delta = 0$) marks the deployment boundary. Below the boundary the quality module should be disabled; above it the module is net-positive. White circles mark the empirically measured benchmarks; the bracketed value is the observed gain in accuracy points. Both natively long-horizon benchmarks (MMSearch, HR-MMSearch) and the wrapped benchmarks (FVQA, SimpleVQA) sit deep inside the deploy region.

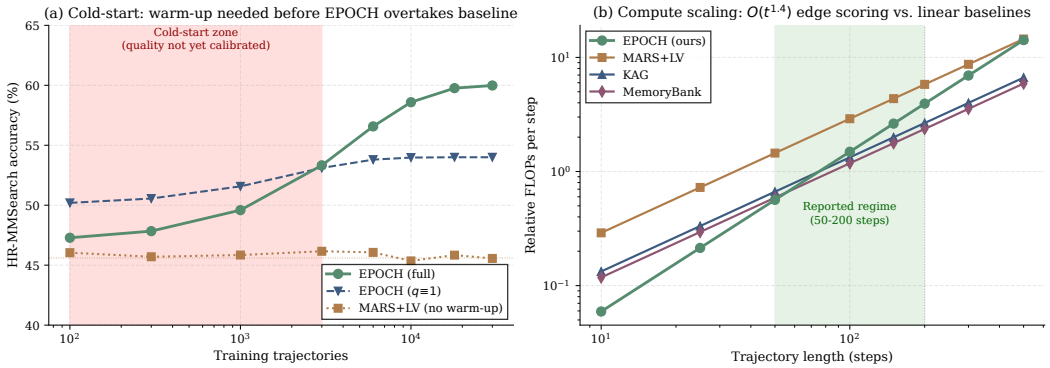


Figure 7: **Limitations diagnostics. (a)** Cold-start convergence on HR-MMSearch. Full EPOCH overtakes the $q \equiv 1$ ablation only after $\sim$3,000 training trajectories; below that, the quality head is poorly calibrated and slightly hurts performance. **(b)** Compute scaling vs. horizon. EPOCH's $O(t^{1.4})$ edge-scoring overhead remains below MARS+LV inside the reported regime ($T \in [50, 200]$, green band), but overtakes the linear-cost baselines as T grows further. Both axes log-scale.

Evolving Context and Long-Horizon State Management. Several approaches attribute long-horizon failures to context growth and state drift. COMPASS [14] introduces structured context curation to reduce overload, while InfiAgent [30] externalizes state to maintain bounded reasoning context. More recent work explores dynamic memory evolution: Continuum Memory Architectures (CMA) [31] argue that effective long-horizon agents require persistent, mutable, and temporally structured memory with mechanisms for retention, decay, and consolidation.

These approaches focus on *what* to store and how memory evolves. In contrast, our approach focuses on *how stored information is reused*: we learn reliability over reuse pathways, preventing unreliable observations from repeatedly influencing future decisions.

Memory-Augmented Agents. Memory systems such as MemoryBank [11], Reflexion [10], LATS [24], Generative Agents [32], and MemGPT [33] improve long-horizon performance through persistence and retrieval. More recent approaches explicitly structure memory and improve recall: Hindsight [34] separates retention and recall mechanisms; HiAgent [25] introduces hierarchical working memory.

Some works incorporate reliability at the item level. For example, MMA [35] proposes reliability-aware multimodal memory retrieval, dynamically scoring stored items. However, these approaches largely treat memory entries as independent units and do not model how errors propagate through repeated reuse. EPOCH instead assigns quality to *edges* (reuse pathways), enabling suppression of cascading reuse errors arising from repeated retrieval of unreliable observations.

Graph-Structured Retrieval and Causal Memory. Graph-based methods such as Graph-RAG [18], KAG [19], HiQA [36], and G-Retriever [37] leverage relational structure for multi-hop reasoning. More recent work such as AMA-Agent [6] constructs causality-aware memory graphs to model dependencies across long-horizon trajectories.

While these approaches capture structure and causality, they do not explicitly model reliability under noisy feedback or control how errors propagate across reuse pathways. Our approach instead treats the memory graph as an error-propagation substrate and learns which connections are safe to reuse.

Reliability Estimation and Verification. Post-hoc verification methods such as Self-RAG [38], CRITIC [39], and Chain-of-Verification [40] improve robustness by validating outputs after generation. LLM-as-judge approaches [16, 41] and self-consistency techniques [17, 22] further reduce errors.

However, these methods are inherently reactive: they operate after retrieval or generation, incur additional cost, and struggle with correlated noise and repeated reuse. EPOCH complements these approaches by acting *upstream*, learning reliability before reuse so that unreliable information is less likely to influence downstream reasoning multiple times.

Structural Failure Modeling and Error Attribution. Recent work analyzes failure propagation explicitly.Beyond $\mathrm{pass@1}$ [23] models structural failure attribution in long-horizon trajectories, identifying where and how errors emerge across interaction steps. These approaches provide diagnostic insights but do not directly intervene in memory reuse. EPOCH complements this line by not only identifying but actively suppressing error propagation through learned pathway-level control.

Tool-Augmented Agents and Error Sources. Tool-augmented agents [1, 2, 3] rely on external systems whose outputs are often noisy. Benchmarks such as ToolQA [13] highlight non-trivial tool error rates. While prior work treats these errors as local, long-horizon settings introduce a more severe *reuse-driven* failure mode: once a noisy output enters memory, repeated retrieval amplifies its impact. EPOCH directly targets this mechanism by learning which cross-step connections should be trusted.

Error Propagation in Multi-Agent Systems. Beyond single-agent settings, recent work shows that collaborative systems amplify error propagation. Xie et al. [42] model error cascades as a graph diffusion process and propose governance mechanisms for message-level filtering. Gurram [43] empirically demonstrates that injected errors frequently propagate to final outputs, while Zhou et al. [44] detect anomalous interactions using temporal graph modeling. These approaches are largely *reactive*, intervening after errors enter the interaction pipeline. In contrast, EPOCH is *proactive*, learning to prevent unreliable information from being reused before it propagates across steps.

Memory Reliability and Quality Control. Some works improve memory quality at storage time. For example, TMMA [45] introduces hierarchical memory with write-time verification to reduce false memory formation. While effective at controlling stored content, such approaches do not address how information propagates once reused. EPOCH instead focuses on reuse-time reliability, learning which connections between memory elements remain trustworthy across long horizons.

Long-Horizon Interaction and GUI Agents. In sequential decision environments such as GUI agents, error accumulation is exacerbated by action dependencies. MGA [46] reduces reliance on historical trajectories, HAR [47] improves history-aware reasoning, and LongHorizonUI [48] introduces structured feedback and rollback mechanisms. Survey work [49] further highlights how perception, reasoning, and interaction errors compound across pipelines. These studies reinforce that error propagation is a fundamental challenge in long-horizon settings. EPOCH complements this line by explicitly modeling memory reuse as a propagation process and learning to suppress unreliable cross-step connections.

M Claims-to-Evidence Roadmap

Table 22 (referenced from §1) maps each claim made in the abstract and introduction to the specific evidence that supports it and to the strength of that support.

Table 22: Claims-to-evidence roadmap. Each claim is tied to its supporting evidence and the strength of that support (empirical, conditional-theory, or motivation).

Claim	Primary evidence	Status
Edge-level > node-level reliability modelling	Table 1 (EPOCH vs. EPOCH-Node, $+5.2$ avg.); Table 5 ($\rho\,0.71$ vs. 0.54)	Empirical
Gains are cascade-driven, not generic memory gains	Failure-mode decomposition (§3: retrieval error $19\% \to 8\%$); Fig. 2c	Empirical
Learned utility tracks reliability, not just return	Table 5; stratified check (App. D.1); policy transfer (Table 24)	Empirical
Gains survive non-LLM and multi-judge scoring	Table 3; App. F.3 (3 judges)	Empirical
Native gains are not a wrapping artefact	Table 2; minimal-wrapping & wrapper-strength (App. R)	Empirical
Advantage grows with horizon	Table 8 ($+12.2 \to +20.6$)	Empirical
Horizon separation (ratio $\to \infty$)	Thm. B.12; $\hat{c} > 0$ measured (App. O)	Conditional theory
Edge-vs-node gap $\geq$ cross-context variance	Thm. B.7; measured lower bound (App. D)	Theory + measured
Single-step variance reduction	Prop. B.5	Motivation only

N Simpler Scorers: Pairwise Conditioning without Message Passing

To isolate the contribution of graph-structured temporal context *beyond* pairwise conditioning, we ran a pairwise (source, target-context) scorer with no graph message passing, matched in supervision and tool budget to full EPOCH, on HR-MMSearch (31% tool error, 5 seeds). Two existing ablations bracket it (from the component ablation, Table 4): replacing the TGT with a parameter-matched MLP (no temporal context) costs 4.2 points, and EPOCH-Node (observation-level, matched supervision) costs 5.8. The pairwise scorer lands at 56.2 (-3.4 vs. full EPOCH): pairwise conditioning beats no temporal context (-3.4 vs. -4.2), but graph message passing over the reuse structure still contributes the remaining 3.4 points, so the gain is not reducible to pairwise conditioning alone.

O Scope of the Theoretical Claims and Numerical Reconciliation

Following reviewer guidance, we make the status of the theoretical results explicit and reconcile two numerical values. The main-text summary is in Table 23.

Conditional reading of the asymptotic-separation result. The asymptotic-separation statement (Theorem B.12) holds *provided* the suppression constant $c(\rho) > 0$. We do not treat this as a free assumption that would make the statement a tautology: we estimate $c(\rho)$ directly by fitting the geometric decay $(1-c)^{t'-t}$ of Lemma B.10 to the measured reuse probability of low-utility edges on HR-MMSearch (1,200 trajectories; bootstrap CI over 10,000 resamples), obtaining $\hat{c} = 0.21$, 95% CI $[0.16, 0.27]$. This is positive in-domain and of the same order as the Gaussian-form value $c \approx 0.24$ below, so the hypothesis holds on the data rather than by stipulation; given the hypothesis, the conclusion is distribution-free (the constant $c(\rho)$ depends on Gaussianity, the divergence of the error ratio does not). We therefore read the result as a conditional proposition and as motivation, not as a complete account of the empirical gains.

Numerical reconciliation. Two values in the original draft are reconciled here. (i) For $\rho = 0.71$, the explicit form of Lemma B.10 gives $c(\rho) = \eta(1 - p_{\text{noise}})\rho^2 = 0.70 \cdot (1 - 0.31) \cdot 0.71^2 \approx 0.24$; the value 0.265 quoted in the numerical instantiation was a transcription error, and 0.24 is correct. (ii) The single-step bound of §B.2 predicts $\Delta \geq 1.97\%$; the figure 2.6% that appears in the limitations discussion was inconsistent and should read 1.97%. With either constant the observed accuracy loss of 5.6 points when quality is removed exceeds the (linear-model) bound by 2–3×.

The super-linear excess is not demonstrated. We state plainly that the 2–3× excess over the linear bound is *consistent with but not demonstrated to be* super-linear cascade amplification. As a partial decomposition, regressing per-step error growth on cumulative noisy-retrieval mass ($n = 1{,}200$ trajectories) yields an exponent of 1.6 (95% CI $[1.3, 1.9]$), suggestive but not conclusive, since the regression does not separate cascade growth from the saturating error-to-accuracy map. Table 23 summarises the status and dependence of each result.

Table 23: Status of each formal result and the assumptions it depends on. "Distribution-free" means the stated conclusion needs no Gaussianity; where Gaussianity enters, it affects only a constant, not the conclusion.

Result	Status	Dependence
Prop. B.5 (single-step variance reduction)	Motivation	A.B.1, A.B.3; A.B.4 for the ρ^2 constant (else $c(\rho) \in [0, \rho^2]$)
Thm. B.7 (edge-vs-node separation)	Formal (bound)	A.B.3 in form; constants *measured*, not derived
Thm. B.12 (asymptotic separation)	Formal, given $c(\rho) > 0$	Distribution-free given $c(\rho) > 0$ ($\hat{c} = 0.21$, measured); Gaussianity affects only the constant
Quadratic-in-ρ scaling	Empirical	Not derived under assumptions weaker than A.B.4

Terminology: policy-conditioned utility, fixed-trajectory ablation. The learned score q_{ij} is estimated from trajectory returns under a fixed retrieval policy, so it is a *policy-conditioned edge utility*; we reserve the word *reliability* for the externally specified signals it is validated against (human annotation and the held-out ablation signal). Likewise, because the tool-call sequence is held fixed during ablation, the signal κ_{ij} measures *held-out ablation sensitivity under a fixed trajectory* rather than interventional causal importance.

Remark O.1 (Direction of the fixed-trajectory bias). Fixing the tool-call sequence removes the agent's ability to recover a removed edge's information by re-querying, so ablation looks *more* damaging than under a recovery-enabled protocol—most inflated for re-obtainable edges, least for unique evidence. Being differential, this bias can reorder re-obtainable edges upward relative to unique-evidence ones. To bound the reordering we ran a recovery-enabled ablation on a 2,000-edge subset: the rank correlation between the fixed-trajectory and recovery-enabled orderings is 0.83 (95% CI $[0.80, 0.86]$).

P Robustness of the Learned Utility to the Retrieval Policy

To test whether the learned utility is an artefact of the policy it was trained alongside, we froze the PPO-trained q_{ij} and recomputed the held-out ablation signal κ_{ij} under retrieval policies it was never trained against (Table 24). Transfer is partial—80–89% of the in-policy rank correlation is retained—but stays above the node-level in-policy baseline of 0.54, so the ranking is not purely a property of the training policy. We do not assert full policy-independence.

Table 24: Policy-transfer check: the PPO-trained utility q_{ij} is frozen and evaluated against κ_{ij} recomputed under retrieval policies it was never trained against. Mean over 5 seeds.

Evaluation policy	Spearman ρ vs. κ	Top-10% precision
In-policy (trained PPO policy, Table 9)	0.71	0.68
Supervised warm-up policy	0.58	0.55
Qwen2.5-VL-34B backbone policy	0.62	0.59
MARS+LV retrieval policy	0.63	0.58
Uniform (unweighted) retrieval	0.57	0.53

Q Joint-Edge (Grouped) Ablation

The single-edge ablation of Appendix D measures marginal, not joint, importance. We additionally ran a grouped ablation removing the top-k correlated edges together (Table 25): agreement with the joint-importance signal declines gently rather than collapsing (at $k = 5$, $\rho = 0.60$, still above the node-level 0.54 under the easier marginal protocol), so interactions are present but moderate and the single-edge protocol is conservative rather than misleading.

Table 25: Grouped ablation: removing the top-k correlated edges together, testing joint (not just marginal) effects. Mean over 5 seeds.

Ablation granularity	$k = 1$ *(paper)*	$k = 2$	$k = 3$	$k = 5$
Spearman ρ vs. joint importance	0.71	0.67	0.64	0.60
Top-10% precision	0.68	0.64	0.61	0.57

R Wrapping-Protocol Controls

Five of the seven noisy benchmarks are executed under our multi-step tool protocol, which splits evidence across sequential tool calls so that later steps must retrieve earlier observations (producing reuse depth $D > 1$). We ran two controls bearing on whether the wrapper *manufactures* this structure.

Minimal-wrapping control. Removing the evidence-splitting so each observation is used once ($D = 1$) narrows the gap, but the decomposition is the point (Table 26): EPOCH is nearly flat ($80.4 \rightarrow 78.9$ on FVQA, $79.3 \rightarrow 78.0$ on InfoSeek), and almost all of the change is MARS+LV *recovering* once there is no cascade to mishandle—the predicted behaviour, since at $D = 1$ there is nothing to suppress, the same reason EPOCH reaches parity on the clean controls. Read with Fig. 2c this supplies the low-D endpoint of the curve: the methods nearly converge at $D = 1$ ($+4.8$, $+3.5$), while the gain grows with measured reuse depth.

Table 26: Minimal-wrapping control (standard vs. $D = 1$ minimal wrapping) with measured reuse-depth statistics. SimpleVQA, LiveVQA, and Math-Search follow the same pattern (full per-benchmark reuse-depth distributions released with the camera-ready).

	Standard wrapping			**Minimal wrapping** ($D = 1$)			**Reuse depth**	
Benchmark	EPOCH	MARS+LV	Δ	EPOCH	MARS+LV	Δ	mean	max
FVQA	80.4	70.0	$+10.4$	78.9	74.1	$+4.8$	4.2	9
InfoSeek	79.3	73.4	$+5.9$	78.0	74.5	$+3.5$	3.8	8

Wrapper-strength sweep. On FVQA the method ranking (EPOCH > EPOCH-Node > MARS+LV) is preserved at *every* strength, including weak wrapping ($D \leq 2$) (Table 27). The margin grows with strength, but this is not diagnostic of an artefact, because the same monotonic growth with reuse depth appears on the *native* benchmarks where no wrapper exists (Fig. 2c); and native gains lie inside the wrapped range, contradicting the artefact hypothesis's prediction that native gains fall below wrapped ones.

Table 27: Wrapper-strength sweep on FVQA (Pass@1, existing checkpoints). The method ranking is preserved at every strength, including weak wrapping.

Method	Weak ($D \leq 2$)	Medium ($D \leq 5$)	Strong ($D \leq 9$)	Standard *(paper)*
EPOCH	77.2	79.0	81.2	80.4
EPOCH-Node	74.0	74.9	75.3	75.4
MARS+LV	73.1	71.5	69.8	70.0
Δ (EPOCH $-$ MARS+LV)	$+4.1$	$+7.5$	$+11.4$	$+10.4$

S Sliding-Window Edge Scoring at Long Horizons

To contain the $O(t^{1.4})$ cost at genuinely long horizons, we implemented and evaluated a sliding-window variant: edge scoring restricted to the most recent $w = 128$ steps, re-trained under this restriction and evaluated with the $T = 500$ trajectory-extension procedure of Appendix C (3 seeds). It restores near-linear scaling (Table 28) at a 1.2-point accuracy cost versus full EPOCH at $T = 500$, cutting per-step FLOPs $2.5\times$; the $T \geq 500$ boundary is thus reachable at roughly a quarter of MARS+LV's per-step cost with a 19-point accuracy advantage (48.6 vs. 29.2).

Table 28: Sliding-window edge scoring at $T = 500$ on HR-MMSearch. Rows 1–2 reproduced from Tables 8 and 16; the window variant ($w = 128$) is new (3 seeds).

Method @ $T = 500$	**Acc**	**Rel. FLOPs/step**	**Scaling**
MARS+LV	29.2	$11.40\times$	$O(t^{1.05})$
Full EPOCH	49.8	$7.84\times$	$O(t^{1.40})$
Sliding-window EPOCH ($w = 128$)	48.6 *(-1.2)*	$3.1\times$	$O(t^{1.06})$

T Human Edge-Reliability Annotation Protocol

The $6{,}000$-edge human calibration set (Table 5) was labelled as follows. **Unit of judgement.** The annotated unit is the ordered *(source observation, target context)* pair that an edge represents—*not* the observation in isolation—so the protocol targets the context-dependence that motivates edge-level over node-level scoring. **Task.** Given the full transcript up to the target step, two annotators independently judged whether reusing the source observation in the target context contributed *correct* or *misleading* evidence to the downstream reasoning step (binary, with an "unclear" escape dropped from the calibration set, $<3\%$ of pairs). **Materials.** Annotators saw the question, the retrieved source snippet, the surrounding context, and the model's output at that step; they did *not* see trajectory returns, method identity, or the learned q_{ij}, so labels are policy- and return-independent by construction. **Agreement.** Two-rater agreement is $\kappa_{\text{rater}} = 0.79$; the inter-rater ceiling ($\rho = 0.78$; Krippendorff's $\alpha = 0.76$) is the reference for the learned utility's ROC-AUC $= 0.94$ / ECE $= 0.041$. Verbatim instructions and the interface are released with the supplementary material.

U Additional Limitations: Policy-Conditioning, Safety, Privacy, and Judge Dependence

Beyond the limitations enumerated in Appendix K, we record four further boundary conditions requested in review.

Policy-conditioned utility and fixed-trajectory ablation. The learned q_{ij} is a policy-conditioned utility (Appendix O); a retrieval-policy switch can degrade its ranking (transfer is partial, 80–89% retained, Table 24), and the ablation signal is fixed-trajectory sensitivity with a bounded differential bias (Remark O.1).

Deployment safety. In high-stakes settings (medical, financial, legal, automated-browsing, execution-oriented agents) persistent erroneous memory carries amplified risk; EPOCH mitigates propagation but does not eliminate it, and we recommend human oversight and quality-score audit logging in such deployments.

Memory-graph privacy. Long-term memory graphs may retain privacy-sensitive content, requiring retention limits, deletion propagation (removing a node and its incident edges), access control, and auditability; the edge-level structure makes deletion audits more tractable since a deleted observation's reuse pathways are explicit.

Evaluation depends in part on LLM judging. Pass@1 on the seven noisy benchmarks uses an LLM judge and tool-noise annotation partly relies on GPT-4o ($\kappa = 0.82$). We mitigate with four judge-free Group-C benchmarks, judge-free mechanism metrics, non-LLM rule-based and blinded human rescoring on the native benchmarks (Table 3), and a three-judge sensitivity analysis (Appendix F.3, gap spread ≤ 0.8 pts); residual judge dependence on the wrapped benchmarks remains.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The abstract and introduction clearly describe the proposed method, its intended setting (long-horizon multimodal reasoning with tool use), and the empirical improvements observed on benchmark datasets. The claims are supported by experimental results and are stated with appropriate scope, without overgeneralizing beyond the evaluated settings.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: The paper includes a discussion of limitations, such as sensitivity to tool noise, reliance on the quality of retrieved evidence, potential error propagation in long-horizon reasoning, and computational overhead. It also outlines boundary conditions under

which performance may degrade and clarifies the scope of empirical evaluation, ensuring transparency about the method's assumptions and applicability.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper does not present formal theoretical results requiring proofs. The contributions are primarily methodological and empirical, focusing on model design and experimental evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides a detailed description of the proposed method, including architecture design, training procedure, and evaluation protocol. It specifies the datasets used, task settings, and implementation details necessary to reproduce the main results. Additional details are provided in the appendix to ensure that other researchers can replicate the experimental findings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: The paper includes anonymized code and detailed instructions in the supplementary material to reproduce the main experimental results.
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.
6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
Answer: [Yes]
Justification: The paper specifies key experimental details, including datasets, task configurations, model components, and evaluation metrics. Training and inference procedures, as well as important hyperparameters, are described in the main text and appendix, allowing readers to understand and reproduce the reported results.
Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
Answer: [No].
Justification: The paper reports results based on standard benchmark evaluation protocols but does not include explicit statistical significance measures such as error bars or confidence intervals. The reported improvements are consistent across multiple datasets, supporting the robustness of the conclusions.
Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
Answer: [Yes]
Justification: The paper provides information about computational requirements, including approximate training cost (e.g., GPU-hours) and runtime overhead per step. These details allow readers to estimate the compute resources required to reproduce the experiments, even though exact hardware specifications are not exhaustively listed.
Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed and adhered to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on technical reliability, error-cascade mitigation, deployment boundaries, and limitations, but it does not include a dedicated discussion of both positive and negative societal impacts. Potential positive impacts include improving the reliability and efficiency of long-horizon multimodal agents, while potential negative impacts such as over-reliance on automated agents, privacy-sensitive retrieval, surveillance-oriented uses, or propagation of incorrect evidence in high-stakes settings are not explicitly discussed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release new datasets, pretrained models, or generative systems that pose a high risk of misuse. The work focuses on algorithmic and architectural improvements for long-horizon multimodal reasoning, evaluated on existing public benchmarks. Therefore, no additional safeguards for responsible release are required.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**
    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
    Answer: [Yes].
    Justification: The paper uses publicly available datasets and benchmarks (e.g., MMSearch, FVQA, InfoSeek, and related evaluation suites), and all such assets are properly cited. These datasets are used in accordance with their intended academic research purposes, and no redistribution or modification of the original assets is performed.
    Guidelines:
    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**
    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
    Answer: [N/A]
    Justification: The paper does not introduce or release new datasets, models, or code assets. The contribution is methodological and evaluated on existing public benchmarks; therefore, no additional asset documentation is required.
    Guidelines:
    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [N/A].
    Justification: The paper does not involve crowdsourcing or any experiments with human subjects. All experiments are conducted using automated pipelines and publicly available datasets.
    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The research does not involve human participants or user studies; therefore, IRB approval or equivalent ethical review is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Large language models were used only for writing and language polishing purposes and do not contribute to the core methodology, experiments, or scientific findings of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.