GRAPH-GUIDED DIFFUSION LANGUAGE MODELS FOR EFFICIENT MULTI-HOP REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Retrieval-Augmented Generation (GraphRAG) has remarkably enhanced large language models on complex reasoning by leveraging structured entity topologies. However, existing frameworks heavily rely on standard autoregressive language models where the nature of inherent sequential generation severely hinders overall inference efficiency. Inspired by Diffusion Language Models (DLMs) that offer massive parallelism via continuous refine-in-parallel decoding, we aim to accelerate GraphRAG in the discrete space. However, it remains non-trivial for two challenges. First, partially denoised drafts are highly dynamic and uncertain, making dynamic graph grounding non-trivial. Second, raw denoising states are inherently noisy and unstable, making synchronous graph retrieval and multi-hop aggregation computationally prohibitive. To this end, we present LADDER, a novel framework that bridges diffusion language modeling with GraphRAG through graph-guided parallel decoding. Specifically, (i) we propose an event-driven self-clocking retrieval, inspired by our key insight that 88% of target entities emerge early in the partially denoised state, leading final commitment by an average of 5.7–9.6 steps. This mechanism dynamically triggers graph retrieval only when the set of graph-linkable entities expands, yielding an asynchronous self-clocking policy that bypasses learned gates or heuristic thresholds. (ii) An incomplete-query graph propagation module is designed to process the newly emerging entity queries using a specialized graph foundation model, continuously aggregating multi-hop evidence to sharpen parallel predictions and accelerate overall decoding convergence. Extensive experiments on three challenging multi-hop QA benchmarks show that LADDER raises average exact match from 39.6% to 45.2% while achieving a $4.1\times$ latency reduction.

1 INTRODUCTION

Graph Retrieval-Augmented Generation (GraphRAG) has remarkably enhanced large language models (LLMs) on complex multi-hop reasoning by leveraging structured entity topologies (Gutirrez et al., 2025a; Luo et al., 2025; Gutirrez et al., 2025b; Edge et al., 2025; Guo et al., 2025; Sarthi et al., 2024). In multi-hop question answering, dependencies are rarely visible in a single lexical query; intermediate factual entities must be established before downstream evidence can be identified. Existing GraphRAG frameworks typically interleave search with autoregressive reasoning (Trivedi et al., 2023; Press et al., 2023; Yao et al., 2023). Under autoregressive decoding, however, these mechanisms inherit a fundamental bottleneck: the search query for a subsequent hop cannot be formulated until all preceding reasoning tokens are sequentially generated, and even confidence-triggered variants (Jiang et al., 2023) still resolve one token position at a time. Consequently, retrieval remains a synchronous operation strictly on the inference critical path, severely hindering overall generation efficiency. Diffusion Language Models (DLMs) (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Nie et al., 2025) introduce a promising paradigm shift by refining all response positions in parallel. Instead of building a committed prefix token-by-token, a DLM's intermediate state acts as a continuous, noisy draft of the entire reasoning trajectory. This structural property suggests a novel role for structured retrieval: dynamic entity-document graphs can guide the DLM throughout its parallel denoising process, connecting emerging entities in the draft to multi-hop evidence and resolving prediction uncertainty on the fly. As evidence sharpens dependent position predictions, more

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

positions clear the sampling threshold in parallel, enabling the trajectory to converge in significantly fewer steps.

However, integrating DLMs for efficient reasoning remains non-trivial due to two fundamental challenges. First, partially denoised drafts are highly dynamic and fluctuate throughout decoding. Retrieving statically from the initial question misses emergent evidence, whereas querying at every denoising step incurs severe latency and overwrites useful conditioning with near-duplicate graph states. Second, raw denoising states are inherently noisy and unstable. Attempting synchronous graph retrieval and multi-hop aggregation directly over raw, unverified token drafts introduces prohibitive computational overheads and risks cascading retrieval errors.

To this end, we present LADDER, a novel framework that bridges diffusion language modeling with GraphRAG through graph-guided parallel decoding. Our framework resolves the above tensions through two key innovations. Specifically, (i) Event-Driven Self-Clocking Retrieval: inspired by our key insight that 88% of target entities emerge early in the partially denoised state, leading final commitment by an average of 5.7–9.6 steps, we propose an event-driven retrieval mechanism. Rather than relying on learned gates or confidence thresholds, LADDER dynamically triggers graph retrieval only when the set of graph-linkable entities expands. This yields an asynchronous self-clocking policy where each emerging entity serves both as a refresh event and as a concrete seed node for graph propagation. (ii) Incomplete-Query Graph Propagation: to process the revisable, incomplete event queries generated during intermediate denoising, an incomplete-query graph propagation module is designed using a specialized graph foundation model (Galkin et al., 2024; Luo et al., 2025; Zhu et al., 2022) (GFM), building on GNN-based retrievers designed to tolerate incomplete or noisy graphs (Mavromatis & Karypis, 2024). We fine-tune the GFM with complementary discovery and retention objectives: one retrieves gold evidence still absent from memory, while the other preserves evidence retrieved at earlier events. An anchor term further ensures that this adaptation transfers zero-shot across unseen evaluation corpora.

Our main technical contributions are summarized as follows:

- **Graph-Guided Diffusion Decoding Paradigm.** We identify the sequential bottleneck of traditional GraphRAG and formulate graph retrieval as a parallel guidance mechanism for DLM denoising trajectories, accelerating decoding while improving answer accuracy.
- **Event-Driven Retrieval Policy.** We establish entity lead time as a strong empirical foundation and introduce a gate-free, self-clocking policy that uses emerging entities as both asynchronous refresh triggers and graph propagation start nodes.
- **Incomplete-Query Graph Foundation Model.** We design an event-conditioned graph propagation model trained with discovery and retention objectives to resolve revisable event queries, achieving robust zero-shot cross-corpus transferability.
- Extensive experiments on three challenging multi-hop QA benchmarks show that LADDER raises average exact match from 39.6% to 45.2% while achieving a $4.1\times$ latency reduction.

2 RELATED WORK

GraphRAG improves multi-hop retrieval by exploiting structured entity–document relations. A common inference protocol is IRCoT-style iterative retrieval: the model retrieves from the question and previously generated reasoning, then produces the next thought (Trivedi et al., 2023; Press et al., 2023; Yao et al., 2023). Under autoregressive decoding, however, a later-hop query cannot be formed until preceding reasoning tokens are generated, placing each retrieval round on the critical path. Graph-based retrievers improve the evidence available at each round: HippoRAG (Gutirrez et al., 2025a) uses personalized PageRank over an entity–document graph; GraphRAG (Edge et al., 2025), LightRAG (Guo et al., 2025), HippoRAG 2 (Gutirrez et al., 2025b), and Youtu-GraphRAG (Dong et al., 2025) improve graph construction or retrieval; and learned retrievers such as GNN-RAG (Mavromatis & Karypis, 2024) and GFM-RAG (Luo et al., 2025; Zhu et al., 2022) improve robustness and transfer. In contrast, LADDER performs graph retrieval within a single diffusion trajectory, replacing sequential thought-level retrieval rounds with entity-triggered graph updates.

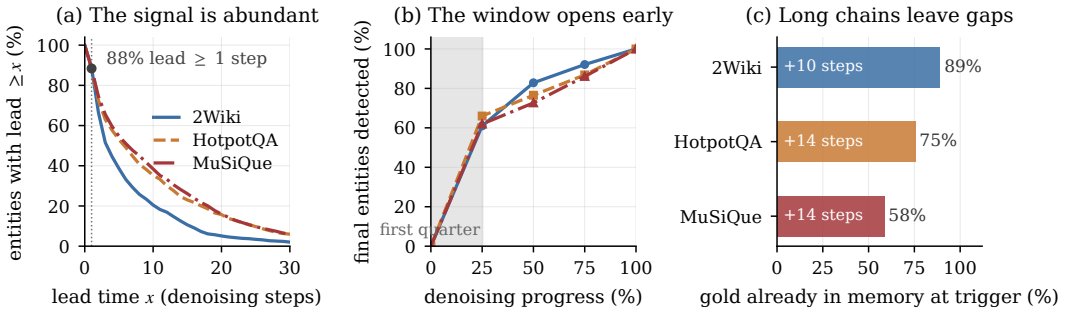

Figure 1: Entities surface before their spans commit, early enough to be useful. (a) Fraction of entities still uncommitted x steps after surfacing in E_t. (b) Fraction of final entities already detected after a given trajectory fraction. (c) Fraction of triggers with all gold documents already in memory; labels give the mean remaining steps. Section 4.4 targets the coverage loss visible here on long reasoning chains.

Diffusion language models. Discrete diffusion over token sequences (Austin et al., 2021; Lou et al., 2024) and its continuous-embedding and masked variants (Li et al., 2022; Sahoo et al., 2024; Nie et al., 2025) decode by refining an entire sequence, so an intermediate state is a noisy draft of the whole response rather than a prefix. We exploit this global draft to couple retrieval scheduling with graph structure: each newly emerging entity in the draft determines both *when* to refresh and *which* seed node to propagate from in the entity–document graph. This coupling exists only once retrieval is graph-structured, and we further sharpen it by training the retriever on the denoising events themselves.

3 Observations: Entities Emerge Before They Are Committed

Our method is motivated by a measurable property of DLM decoding: graph-linkable entities can appear in an intermediate draft before their token spans are committed.

Measurement protocol. We collect denoising trajectories for 1,000 test questions from each of 2WikiMultihopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022). Dream-7B uses a confidence-threshold sampler: at each step, every masked position whose predicted-token confidence exceeds a threshold τ_c is committed and is never revised afterward. At step t, we extract entities from the argmax proxy r_t: committed positions kept as-is, masked positions filled by their current top-1 prediction (Eq. 3). Let $\mathcal{C}_t$ be the positions committed under this rule by step t, and let $\mathcal{J}(e)$ denote the final-response token positions of entity e. We measure

$$t_{\text{emerge}}(e) = \min\{t : e \in \Delta E_t\}, \qquad t_{\text{commit}}(e) = \min\{t : \mathcal{J}(e) \subseteq \mathcal{C}_t\}, \tag{1}$$

$$\text{LeadTime}(e) = t_{\text{commit}}(e) - t_{\text{emerge}}(e). \tag{2}$$

This span-level definition requires all of an entity's token positions to be committed, so it excludes transient proxy guesses without coupling an entity's commitment to unrelated earlier positions.

For a trigger at step t of question i, Figure 1c defines its head start as $H_{i,t} = T_i - t$, where T_i is the final denoising step. It records whether $\mathcal{D}_i^\star \subseteq M_{t-1}$ at that trigger and reports the mean of $H_{i,t}$ over trigger events.

Findings. Figure 1 and Appendix C show that 88% of final entity spans are detected before commitment, with mean lead times of 5.7–9.6 denoising steps. More than 60% of final entities appear in the first quarter of the trajectory, leaving a useful window for evidence retrieval. The window is not uniformly easy to exploit: the fraction of trigger events for which all gold documents are already available falls from 88.9% on 2Wiki to 58.4% on MuSiQue. This motivates an event-conditioned retriever that both discovers missing evidence and preserves evidence found at earlier events. Appendix I illustrates this behavior on one trajectory.

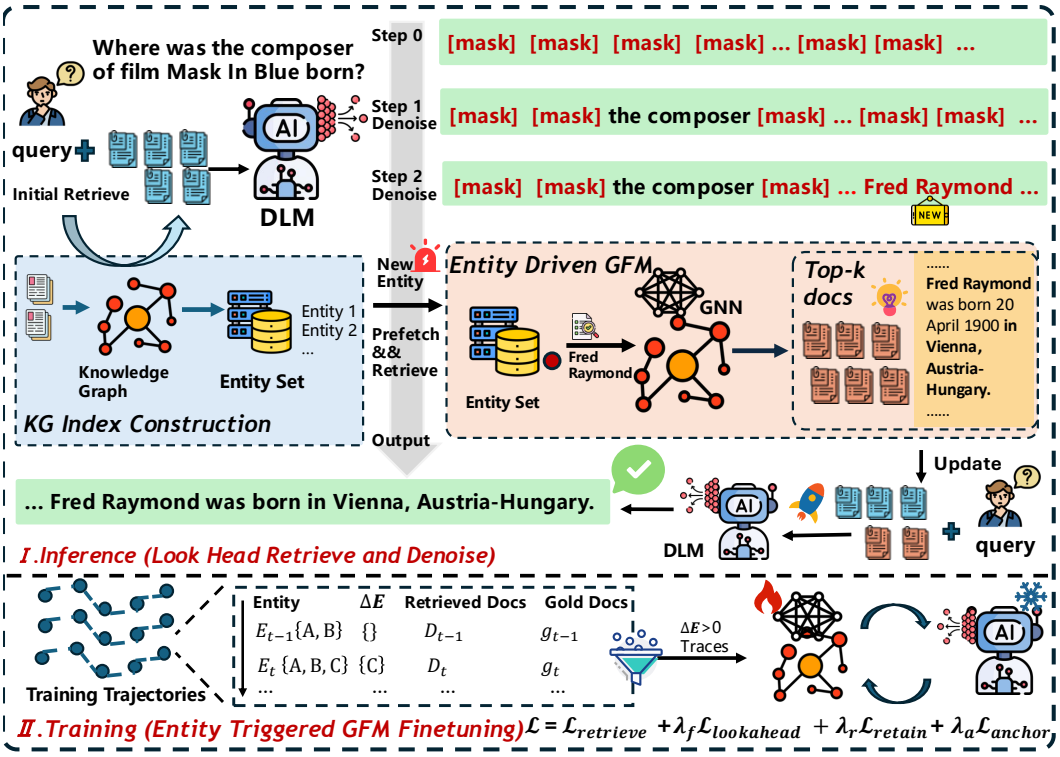

Figure 2: Overview of LADDER. The DLM exposes a partially denoised response state at each step. Newly matched graph entities trigger GFM propagation and retrieve top-K evidence, which conditions subsequent denoising updates. Equation 12 defines the complete retriever-training objective.

4 METHOD

LADDER maintains a diffusion response state, a monotone set of confirmed graph entities, and the evidence retrieved at the most recent event. It uses the response state to expose a retrieval proxy, uses entity emergence to schedule graph propagation, and trains the retriever on the resulting event states.

4.1 DENOISING-TIME RETRIEVAL FORMULATION

Let q denote a question, $\mathcal{G} = (\mathcal{V}, \mathcal{R})$ an entity–document graph, and x_t the discrete diffusion state after denoising step $t \in \{1, \ldots, T\}$. Unlike an autoregressive prefix, x_t contains a partially specified draft of the entire response. We therefore expose a proxy response r_t to the retriever before all of its tokens are committed. For a masked position j, let $p_\theta(\cdot \mid x_t)_j$ be the DLM distribution and $\hat{x}_{t,j} = \arg\max_v p_\theta(v \mid x_t)_j$. Inspired by the idea of using an intermediate diffusion draft as a retrieval signal (Jnger et al., 2026), we form the proxy by retaining committed tokens and, at every masked position, filling in the current argmax guess $\hat{x}_{t,j}$, regardless of its confidence:

$$r_{t,j} = \begin{cases} x_{t,j}, & \text{position } j \text{ already committed,} \\ \hat{x}_{t,j}, & \text{position } j \text{ still masked.} \end{cases} \tag{3}$$

No threshold governs what the proxy exposes: the sampler's confidence threshold decides only which tokens are committed to the final response, and has no bearing on retrieval. The proxy is passed to the entity matcher of Section 4.2 in full, so retrieval is driven solely by which entities it newly contains, the source of the lookahead signal measured in Section 3.

At initialization we retrieve from the question alone. At subsequent steps, the retrieval query is

$$u_0 = q, \qquad u_t = [q \,\|\, r_t], \quad t > 0, \tag{4}$$

where $[\cdot\|\cdot]$ denotes text concatenation. Given retrieved evidence D_t, we rebuild the RAG prompt $P(q, D_t)$ before the next denoising update (Appendix H gives the exact template and output format). At a refresh, we retain the response token identities and commitment mask, but re-tokenize the complete prompt with the updated passages. Dream-7B recomputes RoPE positions over the refreshed full sequence; we use neither fixed context slots nor cross-refresh KV caching. The next denoising update therefore reconditions the retained response state on the new evidence.

4.2 Entity-emergence scheduling

Running graph propagation for every syntactically different proxy is wasteful: most changes in a diffusion state do not change its graph start nodes. We cast scheduling as a discrete graph-state transition. At step 0, we apply NER and embedding-based entity linking to q to obtain initial graph seeds E_0. Thereafter, we match u_t against the closed vocabulary of entity-node names in $\mathcal{G}$. The matcher lowercases and tokenizes the text, then uses a greedy longest-match n-gram lookup (up to six tokens by default); it therefore requires neither an additional language model nor online entity linking. Let

$$\widehat{E}_t = \mathrm{Match}(u_t; \mathcal{V}_{\mathcal{G}}), \qquad \Delta E_t = \widehat{E}_t \setminus E_{t-1}, \qquad E_t = E_{t-1} \cup \widehat{E}_t. \tag{5}$$

The initial linked seeds are included in E_0, and later exact matches are canonical graph node identities. Retrieval is scheduled by

$$a_t = \mathbb{I}\left[|\Delta E_t| \geq 1\right], \qquad t > 0, \tag{6}$$

with an unconditional retrieval at $t = 0$. When $a_t = 0$, we reuse the most recent score vector and evidence set rather than recomputing query features, graph propagation, or document ranking. The policy is self-clocking: the same new entities determine both *when* to refresh and *where* propagation starts. It has no learned gate, confidence calibration, or reward signal.

The entity state is monotone. A speculative mention may disappear as r_t is revised and later recur; removing it from E_t would trigger redundant refreshes. Accumulation makes each canonical entity trigger at most once and preserves it as a graph seed despite surface-form fluctuations.

4.3 Event-conditioned graph retrieval

We instantiate the retriever with GFM-RAG (Luo et al., 2025). At an event, the query encoder produces $h_t = h(u_t)$ and the confirmed entities define a multi-hot graph-start mask $m(E_t)$. An NBFNet-style GNN yields raw entity scores $z_t(e)$. With $A_{ed} \in \{0, 1\}$ denoting the fixed entity–document incidence matrix and $f_e = \sum_d A_{ed}$, the document score is

$$z_t = f_\theta\big(\mathcal{G}, h_t, m(E_t)\big), \qquad s_t(d) = \sum_e \frac{A_{ed}}{f_e} z_t(e), \qquad D_t = \mathrm{TopK}_{d \in \mathcal{D}}\ s_t(d). \tag{7}$$

Here $\mathcal{D} \subset \mathcal{V}$ is the document-node set and entities with $f_e = 0$ have zero weight. During fine-tuning, this dense aggregation scores all documents and is differentiable; inference restores GFM-RAG's pretrained hard Top-20 entity ranker before selecting the final top-K documents. Consequently, retrieval can reach multi-hop evidence with little lexical overlap with u_t. Our default uses cumulative seeding with all entities in E_t; Appendix E compares it with new-seed propagation from ΔE_t. Between events, we reuse the most recent graph scores and evidence set.

The top-K documents at event t form the prompt evidence D_t ($K = 7$ in all main experiments; Appendix F sweeps K). Separately, we define the historical retrieval set $M_t = M_{t-1} \cup D_t$ for training targets and coverage analysis; it need not be identical to the documents in the current prompt.

4.4 Event-conditioned retriever training

A retriever pretrained on complete questions is mismatched to the event states in Equation 4: these queries contain an incomplete, revisable reasoning draft and a newly changed graph seed. We therefore collect *entity-event traces* only from training questions. Each row records $(q, u_t, E_{t-1}, \Delta E_t, E_t, M_{t-1}, D_t, \mathcal{D}_i^\star)$ for a step with $|\Delta E_t| > 0$, where $\mathcal{D}_i^\star$ is the question's gold supporting document set. Query embeddings are cached when traces are collected; the DLM, the graph structure, and the text encoder remain frozen during retriever fine-tuning.

For an event t, define the evidence that has not yet entered memory and the gold evidence already present in memory as

$$\mathcal{F}_t = \mathcal{D}_i^\star \setminus M_{t-1}, \qquad \mathcal{R}_t = \mathcal{D}_i^\star \cap M_{t-1}. \tag{8}$$

The negative pool is the deduplicated non-gold subset of the current hard retrieval candidates and pre-event memory,

$$\mathcal{N}_t = \big(D_t \cup M_{t-1}\big) \setminus \mathcal{D}_i^\star. \tag{9}$$

Thus, the current candidate contribution contains at most $K = 7$ documents; we use neither in-batch nor additional background negatives. For any positive set A, we use the marginal listwise objective

$$\ell_{\text{list}}(A, \mathcal{N}_t) = -\log \frac{\sum_{d \in A} \exp(s_t(d)/\tau)}{\sum_{d \in A \cup \mathcal{N}_t} \exp(s_t(d)/\tau)}. \tag{10}$$

This set-level objective assigns probability mass to the annotated support set; we do not use an average of per-positive losses. The current-evidence and lookahead terms are $\mathcal{L}_{\text{retrieve}} = \ell_{\text{list}}(\mathcal{D}_i^\star, \mathcal{N}_t)$ and $\mathcal{L}_{\text{lookahead}} = \ell_{\text{list}}(\mathcal{F}_t, \mathcal{N}_t)$. The latter explicitly trains an early entity event to retrieve evidence not yet seen by the agent. To avoid demoting correct earlier evidence when a new entity appears, we use a hardest-negative hinge loss,

$$\mathcal{L}_{\text{retain}} = \frac{1}{|\mathcal{R}_t|} \sum_{d^+ \in \mathcal{R}_t} \left[m - s_t(d^+) + \max_{d^- \in \mathcal{N}_t} s_t(d^-) \right]_+ . \tag{11}$$

A term is set to zero when its positive set is empty. Finally, with θ_0 the pretrained GFM parameters, we optimize

$$\mathcal{L} = \mathcal{L}_{\text{retrieve}} + \lambda_f \mathcal{L}_{\text{lookahead}} + \lambda_r \mathcal{L}_{\text{retain}} + \lambda_a \mathcal{L}_{\text{anchor}}, \qquad \mathcal{L}_{\text{anchor}} = \sum_\ell \frac{\|\theta_\ell - \theta_{0,\ell}\|_2^2}{|\theta_\ell|}, \tag{12}$$

where $|\theta_\ell|$ is the number of scalar parameters in tensor ℓ. We fine-tune all GFM GNN parameters using AdamW. Unless otherwise stated, $\lambda_f = 1.0$, $\lambda_r = 0.5$, $\lambda_a = 10^{-5}$, $m = 0.1$, temperature $\tau = 1$, weight decay 0.01, and gradient clipping at 1.0.

5 Experiments

5.1 Experimental setup

Datasets. We evaluate on HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022), using the $1{,}000$-question evaluation sets and corpora adopted by HippoRAG (Gutirrez et al., 2025a;b). Each question is evaluated once, and we construct one entity–document graph per corpus. The lead-time analysis uses the same evaluation questions solely for observation; it is not used for parameter updates, threshold selection, or model selection. Retriever training uses only HotpotQA and 2WikiMultihopQA training traces. MuSiQue is therefore parameter-held-out for retriever optimization and model selection.

Baselines. We compare RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2025), LightRAG (Guo et al., 2025), HippoRAG 2 (Gutirrez et al., 2025b), Youtu-GraphRAG (Dong et al., 2025), and GFM-RAG (Luo et al., 2025). Each baseline uses the same IRCoT-style controller: a round retrieves from the question and prior thoughts, then generates one additional thought. The controller stops after at most five rounds or when the generator emits `###`. Autoregressive variants use Qwen2.5-7B. Dream-7B and Qwen2.5-7B receive matched supervised fine-tuning to produce the same structured reasoning-trace interface (Appendix B). We also evaluate pretrained GFM-RAG with Dream-7B under this multi-round controller. This DLM IRCoT baseline has neither entity-event triggering nor event-conditioned retriever training. LADDER instead refreshes graph evidence within a single denoising trajectory when a new entity emerges.

Metrics, and implementation. Table 2 compares end-to-end systems: autoregressive baselines and GFM-RAG (IRCoT) use thought-level multi-round retrieval, whereas LADDER replaces this sequential loop with event-driven refreshes during parallel denoising. Tables 3 and 8 instead hold a Dream-7B trajectory and a fully event-conditioned GFM retriever fixed, changing only the refresh

Table 1: End-to-end retrieval recall (%). Autoregressive baselines and GFM-RAG (IRCoT) use multi-round thought-level retrieval; LADDER uses event-driven refreshes within one denoising trajectory. **Bold** marks the best value in each column.

	HotpotQA			2WikiMultihopQA			MuSiQue		
Method	R@2	R@5	R@7	R@2	R@5	R@7	R@2	R@5	R@7
Autoregressive agent (Qwen2.5-7B)									
RAPTOR (Sarthi et al., 2024)	30.1	56.8	65.2	43.8	66.1	73.5	17.6	31.9	39.1
LightRAG (Guo et al., 2025)	28.4	54.1	62.8	41.6	63.4	71.1	16.1	29.8	36.9
GraphRAG (Edge et al., 2025)	26.7	51.2	60.1	39.5	60.6	68.4	14.9	27.6	34.7
HippoRAG 2 (Gutirrez et al., 2025b)	32.6	59.6	68.1	46.7	70.8	77.9	21.6	37.6	41.8
Youtu-GraphRAG (Dong et al., 2025)	33.4	61.1	69.6	**47.8**	72.1	79.4	**22.3**	**38.4**	**42.7**
GFM-RAG (Luo et al., 2025)	31.6	58.2	66.7	45.5	69.2	76.5	18.7	33.2	40.3
Diffusion agent (Dream-7B)									
GFM-RAG (IRCoT)	27.5	53.0	61.0	45.3	68.3	73.3	20.4	36.4	40.1
LADDER	**35.0**	**64.4**	**72.9**	47.5	**80.1**	**86.9**	21.1	37.2	41.4

Table 2: End-to-end multi-hop QA performance. Autoregressive baselines and GFM-RAG (IRCoT) use multi-round thought-level retrieval; LADDER uses event-driven refreshes within one denoising trajectory. EM and F1 are percentages; time is per-question wall-clock seconds (↓). **Bold** marks the best value in each column.

	HotpotQA			2WikiMultihopQA			MuSiQue			Average		
Method	EM	F1	Time↓	EM	F1	Time↓	EM	F1	Time↓	EM	F1	Time↓
Autoregressive agent (Qwen2.5-7B)												
RAPTOR	41.0	46.9	6.35	58.3	63.5	6.72	15.3	27.0	7.18	38.2	45.8	6.75
LightRAG	36.5	43.1	7.46	54.2	59.8	7.83	13.8	23.8	8.51	34.8	42.2	7.93
GraphRAG	33.7	40.6	9.72	51.4	56.9	10.16	12.5	21.9	11.24	32.5	39.8	10.37
HippoRAG 2	43.8	50.1	10.86	62.4	67.2	11.42	17.6	30.4	12.67	41.3	49.2	11.65
Youtu-GraphRAG	44.6	51.0	11.58	63.5	68.3	12.31	18.2	**31.1**	14.24	42.1	50.1	12.71
GFM-RAG	42.5	48.7	8.12	61.3	65.2	7.91	16.7	28.8	8.76	40.2	47.6	8.26
Diffusion agent (Dream-7B)												
GFM-RAG (IRCoT)	41.3	47.5	11.54	60.8	64.7	10.54	16.6	29.9	11.83	39.6	47.4	11.30
LADDER	**49.4**	**59.6**	**2.55**	**67.4**	**73.5**	**1.88**	**18.8**	30.7	**3.95**	**45.2**	**54.6**	**2.79**

policy. We report EM, F1, question-averaged Recall@k, per-question wall-clock time, and denoising steps to convergence. Latency is measured on a single NVIDIA H20 GPU with a warmed graph index. Dream-7B uses at most 100 denoising steps and $\tau_c = 0.85$; GFM-RAG-8M uses all-mpnet-base-v2 entity linking and $K = 7$. Matching uses closed-set n-grams of length at most 6 with cumulative seeds.

5.2 MAIN RESULTS

Tables 1 and 2 compare LADDER with six autoregressive graph-retrieval pipelines and the Dream-7B GFM-RAG (IRCoT) bridge baseline. LADDER achieves the best EM on all three datasets and the best recall at nearly every cutoff. On HotpotQA and 2WikiMultihopQA, it improves over the strongest baseline by 4.8 and 3.9 EM points, respectively. Averaged over datasets, it improves EM from 39.6% to 45.2% and F1 from 47.4% to 54.6% over GFM-RAG (IRCoT), while reducing latency from 11.30 to 2.79 s per question (4.1×). It also exceeds the strongest autoregressive pipeline, Youtu-GraphRAG, by 3.1 average EM points while using less than one quarter of its latency.

Figure 3 (top) summarizes this quality–latency trade-off. LADDER occupies the upper-left region because it replaces thought-level retrieval rounds with sparse event-level updates. MuSiQue is held out from retriever optimization and model selection, so it evaluates parameter-held-out transfer. LADDER attains the best MuSiQue EM (18.8% versus 18.2% for Youtu-GraphRAG) at 3.6× lower latency, although it remains 0.4 F1 points below the strongest baseline. This gap is consistent with the lower gold coverage at events for longer reasoning chains (Figure 1c).

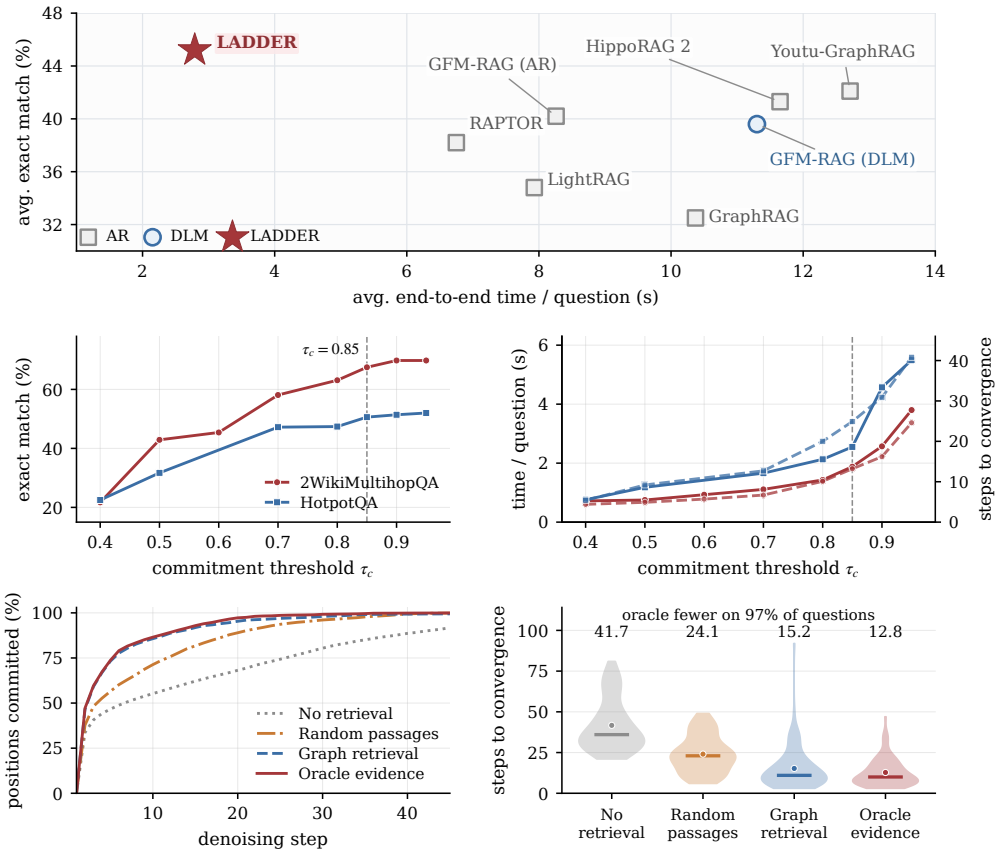

Figure 3: Main results. *Top:* average exact match versus average per-question latency across the three datasets (Table 2), quality increasing upward and latency leftward. *Middle:* quality, latency, and convergence steps as a function of the commitment threshold τ_c (the vertical line marks $\tau_c = 0.85$). *Bottom:* convergence under four evidence conditions on 2WikiMultihopQA, with (left) positions committed per denoising step and (right) the distribution of steps to convergence.

Threshold sensitivity. Figure 3 (middle) sweeps τ_c from 0.4 to 0.95 on HotpotQA and 2WikiMultihopQA. EM plateaus above 0.8, whereas latency and convergence steps increase with the threshold. We therefore use one shared operating point, $\tau_c = 0.85$. Retrieval quality changes by less than two R@7 points over the sweep because entity events, rather than the number of denoising steps, determine refreshes.

5.3 Evidence quality accelerates denoising

A DLM commits masked positions in parallel once their confidence exceeds the sampler threshold. Better evidence should resolve dependencies, commit more positions per step, and shorten the trajectory. We test this prediction by fixing the model, questions, and decoding hyperparameters while varying only the K prompt passages: no evidence, random corpus passages, graph retrieval, or oracle evidence. Oracle prompts place gold documents first and pad the remaining context with retrieved passages.

Figure 3 (bottom) confirms a monotone effect: convergence requires 41.7 steps without evidence, 24.1 with random passages, 15.2 with graph retrieval, and 12.8 with oracle evidence. Oracle evidence uses fewer steps than no retrieval on 97% of questions. Random context removes 42% of steps, and graph retrieval removes a further 37% relative to random context at fixed K. The same ordering holds on HotpotQA and MuSiQue, although the gap narrows for longer chains (Appendix G). Thus, the end-to-end latency gain reflects faster denoising convergence, not merely fewer retrieval calls.

Table 3: (a) Single-trajectory refresh schedules, averaged over three datasets.

Schedule	Time(s)↓	R@7	EM
Retrieve-once	**2.55**	55.6	37.9
Every 5 steps	2.98	62.9	42.6
Per-step	4.38	65.8	44.5
Random (15%)	2.81	59.7	40.4
LADDER	2.79	**67.1**	**45.2**

Table 4: (b) Training objectives, averaged over three datasets.

Objective	R@7	R_{disc}	EM
$\mathcal{L}_{\text{retrieve}}$ only	62.0	39.7	41.9
$+ \mathcal{L}_{\text{lookahead}}$	65.4	49.8	44.2
Full $- \mathcal{L}_{\text{anchor}}$	65.4	50.3	44.2
Full (LADDER)	**67.1**	**52.1**	**45.2**

5.4 ABLATIONS

Table 3 fixes the generator, retriever, and evidence budget while varying the refresh policy. At comparable latency, LADDER improves over RANDOM by 7.4 R@7 and 4.8 EM. Per-step retrieval is 57% slower and remains worse, showing that repeated retrieval without a changed entity set adds little useful evidence. The same ordering holds on MuSiQue (Table 8).

Table 4 fixes the LADDER schedule and varies the retriever objective. Discovery recall, R_{disc}, measures the fraction of gold documents absent from memory that an event retrieves. $\mathcal{L}_{\text{lookahead}}$ adds 10.1 discovery points by supervising missing evidence. Adding $\mathcal{L}_{\text{retain}}$ without $\mathcal{L}_{\text{anchor}}$ leaves average R@7 and EM unchanged and degrades MuSiQue transfer (Table 9). The final $+1.7$ R@7 and $+1.0$ EM arise after adding $\mathcal{L}_{\text{anchor}}$, indicating that anchoring stabilizes event-conditioned retention around the pretrained transferable representation.

6 CONCLUSION

We presented LADDER, a graph-guided diffusion decoding framework for efficient multi-hop reasoning. By triggering retrieval when graph-linkable entities emerge in partially denoised drafts, LADDER replaces sequential thought-level retrieval with sparse graph updates within a single diffusion trajectory. Event-conditioned GFM training further enables robust retrieval from incomplete and revisable queries. Across three multi-hop QA benchmarks, LADDER improves average EM from 39.6% to 45.2% while achieving a $4.1\times$ end-to-end speedup, demonstrating that graph retrieval can both improve reasoning quality and accelerate diffusion decoding.

7 LIMITATIONS

LADDER uses a closed-set entity matcher over a fixed corpus graph, and may miss paraphrases, pronouns, or entities absent from the graph. We evaluate only English multi-hop QA with Dream-7B; however, the event-driven policy is orthogonal to DLM architectures and decoding schemes, requiring only revisable intermediate drafts. Empirical validation across additional backbones, languages, and dynamic graphs remains future work.

8 ETHICS STATEMENT

This work is basic research on retrieval-augmented language models using publicly available multi-hop QA benchmarks; it involves no human subjects or private data collection. A potential limitation is that learned retrieval and generation components may inherit factual errors, biases, or robustness issues from their underlying corpora and models.

9 AI USE STATEMENT

We used generative AI tools to improve the clarity and polish of the writing and to assist with retrieval and discovery of related literature. We did not use generative AI for research ideation or execution, drafting sections of the paper, generating synthetic data, or proving mathematical claims.

All AI-assisted suggestions and retrieved references were reviewed and verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Junnan Dong, Siyu An, Yifei Yu, Qian-Wen Zhang, Linhao Luo, Xiao Huang, Yunsheng Wu, Di Yin, and Xing Sun. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning, 2025. URL https://arxiv.org/abs/2508.19855.

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, 2025. URL https://arxiv.org/abs/2404.16130.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning, 2024. URL https://arxiv.org/abs/2310.04562.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jimnez Gutirrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models, 2025a. URL https://arxiv.org/abs/2405.14831.

Bernal Jimnez Gutirrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models, 2025b. URL https://arxiv.org/abs/2502.14802.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

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

Paul Jnger, Justin Lovelace, Linxi Zhao, Dongyoung Go, and Kilian Q. Weinberger. Self-augmenting retrieval for diffusion language models, 2026. URL https://arxiv.org/abs/2606.06474.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343. Curran Associates, Inc., 2022. doi: 10.52202/068431-0313. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1be5bc25d50895ee656b8c2d9eb89d6a-Paper-Conference.pdf.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024. URL https://arxiv.org/abs/2310.16834.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: Graph foundation model for retrieval augmented generation, 2025. URL https://arxiv.org/abs/2502.01113.

Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for large language model reasoning, 2024. URL https://arxiv.org/abs/2405.20139.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/eb0b13cc515724ab8015bc978fdde0ad-Paper-Conference.pdf.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10, 2022.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction, 2022. URL https://arxiv.org/abs/2106.06935.

A GNN FORWARD AND DIFFERENTIABLE DOCUMENT SCORING

Inputs and seed initialization. We retain the GFM-RAG NBFNet-style reasoning architecture and fine-tune its reasoning parameters on event traces. For an event query u_t, the frozen text encoder produces $h_t \in \mathbb{R}^{d_h}$. Learned projections map the query and fixed relation attributes to the NBFNet hidden dimension d,

$$q_t = \phi_q(h_t) \in \mathbb{R}^d, \qquad R = \phi_r(\text{rel_attr}) \in \mathbb{R}^{|\mathcal{R}| \times d}. \tag{13}$$

Let $m(E_t) \in \{0,1\}^{|\mathcal{V}_E|}$ be the cumulative multi-hot seed mask over entity nodes. The initial node state is the outer product

$$X_t^{(0)} = m(E_t) \otimes q_t, \qquad X_{t,v}^{(0)} = m_v(E_t) q_t, \tag{14}$$

so every confirmed seed receives the same event-query representation and all non-seed entities receive the zero boundary state.

Relation-aware propagation. The query-dependent NBFNet reasoner applies L Bellman–Ford-style graph layers while retaining the original boundary state at every layer. Writing $H_t^{(0)} = X_t^{(0)}$, the recurrence is

$$H_t^{(\ell)} = \mathcal{B}_\ell\big(H_t^{(\ell-1)}, X_t^{(0)}, q_t, R, \mathcal{G}\big), \qquad \ell = 1, \ldots, L, \tag{15}$$

where $\mathcal{B}_\ell$ performs relation-conditioned message passing over the entity graph, aggregates incoming messages, and combines them with the boundary condition. Residual connections are applied when input and output dimensions match. The readout concatenates the configured hidden state(s) with q_t and maps them to scalar entity scores,

$$z_t = \rho_\theta\big(H_t^{(1:L)}, q_t\big) \in \mathbb{R}^{|\mathcal{V}_E|}. \tag{16}$$

This formulation makes the event mask affect propagation only through $X_t^{(0)}$; the graph topology and relation attributes are fixed across events.

Document readout and optimization. The fixed entity–document incidence matrix maps z_t to the dense document scores in Equation 7. During training, every document receives a differentiable score before the listwise and hinge losses are evaluated. Thus, gradients from $\mathcal{L}_{\text{retrieve}}$, $\mathcal{L}_{\text{lookahead}}$, and $\mathcal{L}_{\text{retain}}$ flow through document aggregation, entity readout, NBFNet layers, and the query and relation projections. At inference, we restore GFM-RAG's pretrained hard Top-20 entity ranker before selecting the final top-K documents. The graph structure, entity–document incidence matrix, text encoder, scheduler, and entity matcher remain frozen; only the GFM reasoning model and its projection/readout parameters are optimized.

B GENERATOR FINE-TUNING

LADDER requires intermediate drafts to expose interpretable entities, so we fine-tune Dream-7B to produce structured multi-hop reasoning traces rather than relying on prompting alone. Qwen2.5-7B receives identical supervision data, a matching output format, and the same training configuration, so both generators share a matched reasoning-trace interface; differences reported in the main paper are therefore attributable to retrieval control rather than generator capability. Each training example pairs a question and its supporting documents with a step-by-step trace ending in `### [answer]`, matching the shared IRCoT-style output format.

Output-mode analysis. Following the base-versus-fine-tuned output audit of SARDI (Jnger et al., 2026), we categorize outputs on a 500-question 2WikiMultihopQA subset under static $K = 7$ retrieval (the RETRIEVE-ONCE condition in Table 8) as correct with a structured trace, correct without one, or wrong (including format failures such as a missing `###` marker).

Base Dream-7B rarely emits a structured trace under prompting alone: it is correct with a trace on only 1% of questions (with another 23% correct but trace-free) and wrong otherwise. Base Qwen2.5-7B reasons far more often (29% correct with a trace) but still falls well short of either fine-tuned model. After fine-tuning, both generators reason in nearly every output (58% and 59%

correct with a trace for Dream-7B and Qwen2.5-7B, respectively, with the remainder wrong and no trace-free correct answers) and land within one point of each other, consistent with the 57.6% static-retrieval EM in Table 8. This supports treating fine-tuning as a matched interface rather than a source of asymmetric capability between generators.

Fine-tuning configuration. We fine-tune Dream-7B and Qwen2.5-7B with an identical configuration: 3 training epochs, learning rate 2×10^{-6}, global batch size 256, micro batch size 16 per GPU, gradient accumulation steps 2, maximum sequence length $2{,}048$, and the AdamW optimizer. Training is performed on eight NVIDIA H20 GPUs and inference on a single NVIDIA H20 GPU.

C Entity lead time statistics

Table 5 reports the per-dataset statistics underlying the lead-time claim in Section 3 and Figure 1.

Table 5: Entity span-level lead time, in denoising steps, over $1{,}000$ test questions per dataset. An entity is committed when all of its token positions are independently finalised by the sampler. Reproduced across nine shards (three per dataset), where the Lead > 0 rate stays within 85.8%–88.8%.

Dataset	#Entities	Mean	Median	P90	Lead > 0
2WikiMultihopQA	1,892	5.71	3.0	15	**88.8%**
HotpotQA	3,090	9.35	5.0	24	**88.0%**
MuSiQue	4,524	9.62	6.0	24	**88.5%**

Median lead time is shorter than the mean on all three datasets, indicating a right-skewed distribution: most entities surface only a few steps before commitment, while a minority emerge much earlier, as reflected in the P90 column. The consistently high Lead > 0 rate across shards indicates that early emergence is a stable property of the sampler rather than an artifact of a particular evaluation split.

D Trigger quality of the entity matcher

We report how often an entity event yields a useful retrieval on all three benchmarks. A retrieval is a real GNN forward (a cache hit fires no retrieval when no new entity appears); we call it *useful* when it increases the number of gold documents in the top-K list. Since a retrieved gold document is retained, we measure the rate over triggers fired before full gold coverage, where an increase is still possible.

Table 6: Trigger quality across benchmarks. *Triggers/q* is the mean number of real graph retrievals per question; *Useful-trigger rate* is the fraction, among triggers before full gold coverage, that increase the number of gold documents in the top-K list.

Dataset	Triggers/q	Useful-trigger rate
HotpotQA	3.17	53.3%
2WikiMultihopQA	2.07	79.0%
MuSiQue	4.35	42.6%

The matcher remains sparse, with 2.07–4.35 retrievals per question. Its useful-trigger rate is highest on 2WikiMultihopQA and lowest on MuSiQue, consistent with longer reasoning chains spreading the remaining evidence across more retrieval events.

E Per-dataset ablation results

Tables 3 and 4 report averages for readability. Table 8 and Table 9 give the corresponding per-dataset numbers, from which those averages are computed. Table 7 additionally reports the seed-policy ablation referenced in Section 4.3.

Table 7: Cumulative versus new-only seeding of the GNN start mask, per dataset, with the trigger and all other settings fixed. Cumulative (LADDER's default) seeds every entity confirmed so far, E_t; new-seed uses only the entities that triggered the current event, ΔE_t.

	HotpotQA		2WikiMultihopQA		MuSiQue	
Seed policy	R@7	EM	R@7	EM	R@7	EM
New-seed only (ΔE_t)	69.1	46.8	82.7	63.9	39.2	17.1
Cumulative (E_t)	**72.9**	**49.4**	**86.9**	**67.4**	**41.4**	**18.8**

Cumulative seeding outperforms new-seed propagation on every dataset, most notably on 2WikiMultihopQA ($+4.2$ R@7, $+3.5$ EM), confirming that carrying earlier confirmed entities forward as graph seeds provides persistent value beyond the entities that triggered the current event.

Table 8: Single-trajectory refresh schedules under the same Dream-7B backbone and full event-conditioned GFM retriever. Time is per-question end-to-end wall-clock seconds (↓). These ablations isolate refresh policy and use a different execution unit from the multi-round IRCoT baseline in Table 2. **Bold** marks the best value in each column.

	HotpotQA				2WikiMultihopQA				MuSiQue			
Schedule	Time↓	R@7	EM	F1	Time↓	R@7	EM	F1	Time↓	R@7	EM	F1
Retrieve-once ($t = 0$)	**2.28**	61.9	42.1	51.8	**1.75**	71.2	57.6	62.4	**3.62**	33.6	13.9	24.1
Fixed, every 10 steps	2.41	66.8	45.4	55.1	1.85	76.8	61.2	66.3	3.80	36.2	15.4	26.3
Fixed, every 5 steps	2.67	69.7	47.2	57.0	2.05	80.5	63.8	69.1	4.21	38.5	16.8	28.2
Per-step	3.94	72.4	48.8	58.9	3.02	84.1	66.2	72.0	6.18	41.0	18.4	30.3
Random, firing rate 15%	2.52	66.1	44.8	54.5	1.94	76.1	60.6	65.7	3.97	36.8	15.9	26.9
Conf.-drop (Jiang et al., 2023)	2.49	68.9	46.5	56.2	1.91	81.7	64.9	70.2	3.88	38.9	17.2	28.6
LADDER	2.55	**72.9**	**49.4**	**59.6**	1.88	**86.9**	**67.4**	**73.5**	3.95	**41.4**	**18.8**	**30.7**

Table 9: Training objectives, per dataset. **Bold** marks the best value in each column.

		HotpotQA				2WikiMultihopQA				MuSiQue			
	Objective	R@7	R_{disc}	EM	F1	R@7	R_{disc}	EM	F1	R@7	R_{disc}	EM	F1
(1)	$\mathcal{L}_{\text{retrieve}}$ only	66.4	42.7	45.2	54.0	78.9	50.2	63.4	68.5	40.6	26.1	17.2	30.0
(2)	$+\mathcal{L}_{\text{lookahead}}$	70.8	53.1	48.1	57.8	84.2	62.8	66.1	71.8	41.1	33.4	18.3	30.4
(3)	Full (LADDER)	**72.9**	**55.8**	**49.4**	**59.6**	**86.9**	**65.7**	**67.4**	**73.5**	**41.4**	**34.9**	**18.8**	**30.7**
(4)	Full $-\mathcal{L}_{\text{anchor}}$	71.2	54.2	48.3	58.1	85.1	64.0	66.3	72.1	39.8	32.7	17.9	29.6

F RETRIEVAL-BUDGET SENSITIVITY

Section 4.3 fixes the retrieval budget at $K = 7$ for all main-text experiments. Table 10 sweeps $K \in \{5, 7, 10, 15\}$ on 2WikiMultihopQA and MuSiQue, holding the generator, retriever checkpoint, and event-triggering policy fixed and varying only the number of top-ranked documents placed in the prompt at each retrieval event.

Recall improves monotonically with K on both datasets, as expected: a larger budget can only add documents to the retrieved set. EM and F1 do not track recall monotonically, however. On 2WikiMultihopQA, $K = 10$ attains the highest EM and F1 in the sweep, slightly above the $K = 7$ operating point, while $K = 15$ EM falls back despite higher recall, indicating that additional lower-ranked passages can dilute the prompt with distracting context. On MuSiQue, EM and F1 increase monotonically with K up to 15, consistent with its longer reasoning chains and lower per-event gold coverage (Section 3), which leave more headroom for extra evidence to help. In both cases, however, latency grows steeply with K (by $1.9\times$ from $K = 7$ to $K = 15$ on 2WikiMultihopQA and MuSiQue alike), while the trigger count *Triggers/q* stays nearly flat, confirming that the extra cost

Table 10: Sensitivity of LADDER to the retrieval budget K on 2WikiMultihopQA and MuSiQue, with the generator, retriever, and trigger policy held fixed. EM and F1 are percentages; R@K is the final-state recall at the evaluated budget; *Triggers/q* is the mean number of real graph retrievals per question; latency is per-question wall-clock seconds (↓).

Dataset	K	EM	F1	R@K (final)	Triggers/q	Time↓
2WikiMultihopQA	5	59.0	64.6	80.1	2.07	1.56
	7	67.4	73.5	86.9	2.07	1.88
	10	73.0	76.3	92.1	2.08	2.70
	15	71.0	75.7	93.9	2.09	3.55
MuSiQue	5	15.0	25.1	37.2	4.14	3.17
	7	18.8	30.7	41.4	4.35	3.95
	10	19.0	28.8	41.5	4.39	5.80
	15	23.0	32.0	47.2	4.53	7.55

comes from scoring and prompting more documents per event rather than from additional retrieval events. We use $K = 7$ as a shared latency–quality operating point: compared with the EM-optimal budgets, it reduces latency from 2.70 to 1.88 s on 2WikiMultihopQA and from 7.55 to 3.95 s on MuSiQue, at costs of 5.6 and 4.2 EM points, respectively. K was fixed without MuSiQue model selection; its sweep is post-hoc.

G CONVERGENCE ON THE REMAINING DATASETS

Section 5.3 reports the evidence-quality experiment on 2WikiMultihopQA. Table 11 repeats it on HotpotQA and MuSiQue under identical conditions: same checkpoint, same question set per dataset, same decoding hyper-parameters, and only the $K = 7$ passages in the prompt varying. Figure 4 plots the HotpotQA curves in the same format as the main text.

Table 11: Steps to convergence by evidence condition. *Paired* is the share of questions on which oracle evidence converges in strictly fewer steps than no retrieval.

Dataset	Evidence	Mean	SD	Median	Paired
2WikiMultihopQA	No retrieval	41.70	15.68	36.0	—
	Random passages	24.05	11.30	23.0	—
	Graph retrieval	15.24	12.57	11.0	—
	Oracle evidence	**12.76**	8.43	10.0	97%
HotpotQA	No retrieval	37.91	11.37	37.0	—
	Random passages	31.00	13.54	31.0	—
	Graph retrieval	24.30	14.73	22.0	—
	Oracle evidence	**20.21**	12.99	17.0	85%
MuSiQue	No retrieval	42.39	11.08	41.0	—
	Random passages	37.36	13.81	36.0	—
	Graph retrieval	34.05	17.21	32.5	—
	Oracle evidence	**33.22**	18.51	32.0	74%

The ordering of the four conditions is the same on all three datasets, but the size of the effect is not, and it tracks the length of the reasoning chain. Oracle evidence cuts the trajectory to 31% of its unretrieved length on 2WikiMultihopQA, to 53% on HotpotQA, and only to 78% on MuSiQue, where the gap between retrieved and oracle evidence also narrows to 0.8 steps. This is consistent with Figure 1c: MuSiQue questions require the longest chains and have the lowest fraction of trigger events at which the gold documents are already available, so a single injection of evidence resolves a smaller share of the draft's dependencies and more of the decoding work is irreducible. The mechanism is therefore strongest exactly where multi-hop evidence is most self-contained, and weakest where dependencies are deepest.

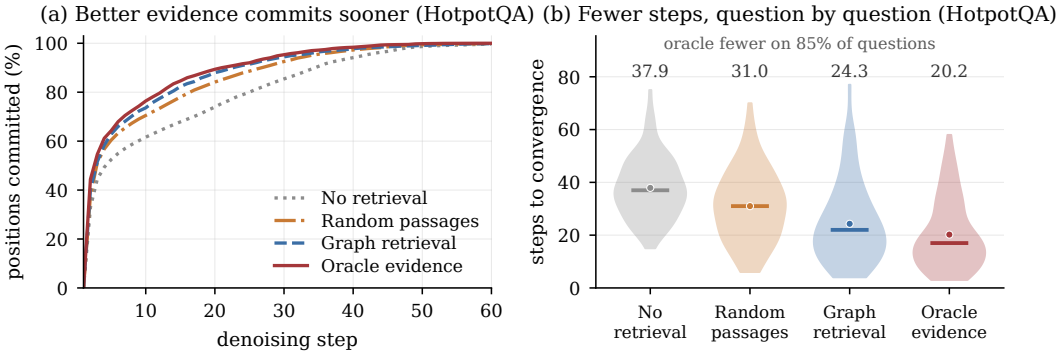

Figure 4: Convergence on HotpotQA, in the format of Figure 3.

H PROMPT TEMPLATES

RAG input format. All models receive input in the following format:

```
Use ONLY the provided facts to answer the question.
Think step-by-step, then provide the final answer
after the "###" marker.
Question:
{question}
Facts:
{facts}
```

where {question} is the input question and {facts} contains the concatenated retrieved passages.

Expected output format. Models are trained to produce outputs in the following format:

```
Step 1:  [First reasoning step grounded in documents]
Step 2:  [Second reasoning step]
...
### [final answer]
```

This format serves two purposes: it enables extraction of intermediate entities from partial reasoning traces for self-augmenting retrieval, and it provides a consistent extraction point (###) for evaluation.

I CASE STUDY

Table 12 traces every denoising step of a single question from a motivation-collection trajectory (Section 3), to make the entity-emergence signal, the caching it enables, and its effect on retrieved evidence concrete rather than aggregate. The question, *"What larger educational institution is the university where Charlotte Brewer went, a part of?"*, is a bridge question: the second hop, which college Charlotte Brewer attended, is absent from the query itself, so the two gold documents (CHARLOTTE BREWER and HERTFORD COLLEGE, OXFORD) cannot both be retrieved from the question alone. At step 0 the initial query retrieves one of the two.

At step 1 the proxy has drifted to "...is at H Soulsford College, Oxford...", a plausible-sounding but fabricated college name. The entity matcher extracts *college*, *Oxford* and *University* from this text. These are valid but generic graph nodes, so they form a genuine event ($|\Delta E_t| \geq 1$) and refresh retrieval, yet their diffuse propagation does not materially change the retrieved set and gold coverage remains $1/2$. The fabricated string "H Soulsford College" itself is not matched: the closed-vocabulary matcher (Section 4.2) admits only canonical graph nodes, preventing nonexistent entities from entering the seed mask. Between steps 2 and 6 the proxy's second half continues to oscillate, reading "professor offord College" at step 2, then stable but wrong through step 6. During this span,

Table 12: Case study: *"What larger educational institution is the university where Charlotte Brewer went, a part of?"* Gold documents: CHARLOTTE BREWER and HERTFORD COLLEGE, OXFORD. Bold tokens in the denoising state are independently committed positions; the remainder is the proxy's argmax fill of still-masked positions. GNN marks a forward pass; ✓ marks a scheduled retrieval call served by cached graph scores; and ∅ marks a step where the unchanged query skips the retrieval call. Gold is the number of gold documents in the retrieved top-K out of the total.

Step	Event	Retrieval	Denoising state u_t	Gold
0	—	fwd	*(question only)*	$1/2$
1	{college, Oxford, University}	fwd	**Charlotte Brewer** is at H Soulsford College, Oxford. (2): …	$1/2$
2	—	✓	**Charlotte Brewer** is at professor offord College, Oxford. (2): …	$1/2$
3	—	✓	*(wording drifts further; entity set unchanged)*	$1/2$
4–5	—	∅	*(proxy text identical to step 3)*	$1/2$
6	—	✓	**Charlotte Brewer** is at professor offord College, Oxford. (2): … *(still wrong)*	$1/2$
7	{Hertford College}	fwd	**Charlotte Brewer** is at Hertford College, Oxford. (2): …ert College Oxford …	$\mathbf{2/2}$
8	—	✓	**Charlotte Brewer is** College. (2): Hert College a constituent college of the University ### Oxford	$2/2$
9	—	✓	**Charlotte Brewer is at Hertford College, Oxford. (2): College a constituent college of the University ###**	$2/2$
10	—	∅	**Charlotte Brewer is at Hertford College, Oxford. (2): College a constituent college of the University ### Oxford**	$2/2$

no new entity is matched, so the seed mask never changes and no GNN forward pass is issued; two levels of caching absorb the five idle steps. When the proxy text itself is byte-identical to the previous step (steps 4 and 5), the query cache returns the previous passages immediately, at negligible cost. When the text has drifted but the entity set has not (steps 2, 3 and 6), the closed-set matcher returns the cached graph-score vector; neither query encoding nor document re-scoring is performed. These cache-served calls take 0.0006–0.003 s, versus 0.055 s for the step-1 event that changes the seed mask. Across the trajectory only 3 of the 11 steps issue a GNN forward pass, while the remaining 8 are served from cache or skipped outright, as anticipated in Section 4.3.

At step 7 the masked positions resolve to "…is at Hertford College, Oxford…", and the matcher extracts HERTFORD COLLEGE, the exact title of the missing gold document. The entity is legible in the proxy two full steps before all of its token positions are committed, illustrating the span-level lead-time distribution in Figure 1a. Because the entity is a real graph node, this event does change the retrieved set: the GNN performs a forward pass (its third and last in this trajectory, after the unconditional retrieval at step 0 and the false-start event at step 1) seeded with the accumulated entities {*college*, *Oxford*, *University*, CHARLOTTE BREWER, HERTFORD COLLEGE}, and HERTFORD COLLEGE, OXFORD enters the retrieved set at rank 2. Gold coverage reaches $2/2$ at this step and remains there for the rest of denoising. All token positions of "Hertford College" are committed at step 9, and the trajectory terminates at step 10 with the correct final answer, *University of Oxford*.

This sparsity (only 3 of 11 steps issuing a GNN forward pass) is the mechanism behind the 15% average firing rate reported throughout the paper. The one useful event, at step 7, is also the one that could not have been triggered by monitoring only committed positions, since the span "Hertford College" is not fully committed until two steps later; a scheme that waits for committed text would miss this window entirely, and a scheme that fires on any token-level change in the proxy would instead fire on every one of steps 2, 3 and 6, where the draft's wording shifts but no new entity, and hence no new evidence, is available.