
Latent Abstraction for Retrieval-Augmented Generation

Ha Lan N. T.^{*1} Minh-Anh Nguyen^{*1} Dung D. Le¹

Abstract

Retrieval-Augmented Generation (RAG) has become a standard approach for enhancing large language models (LLMs) with external knowledge, mitigating hallucinations, and improving factuality. However, existing systems rely on generating natural language queries at each hop and maintaining a strict architectural separation between retriever and generator, preventing them from leveraging the full representational capacity of the LLM. We propose **LAnR** (Latent Abstraction for RAG), a unified framework in which a single LLM jointly performs encoding, retrieval, and generation entirely within its own latent space. Rather than generating textual queries, LAnR produces dense retrieval vectors from the hidden states of a designated `[PRED]` token and uses them to match against encoded document representations from the same model. Furthermore, LAnR adaptively decides when sufficient evidence has been retrieved using a lightweight MLP control head over those same hidden states, eliminating explicit token-level stopping reasoning. Extensive experiments on five QA benchmarks spanning single-hop and multi-hop settings demonstrate that LAnR outperforms existing RAG methods, while achieving improved inference efficiency through reduced number of retrieval calls and tighter model integration.

1. Introduction

Retrieval-Augmented Generation (RAG) has emerged as a standard paradigm for enhancing LLMs with external knowledge, improving factuality, and mitigating hallucinations (Asai et al., 2023a; Ge et al., 2023; Ram et al., 2023). By retrieving relevant documents from external corpora

and conditioning generation on this information, RAG systems enable LLMs to access up-to-date and domain-specific knowledge beyond their parametric memory. Despite these advantages, existing RAG frameworks exhibit several limitations. First, they rely heavily on explicit text-based retrieval, where the model must generate natural language queries to interact with a separate retrieval module. Second, they enforce a strict architectural separation between the retriever and the generator, often requiring independently trained components or additional fine-tuning via reinforcement learning (RL) or supervised fine-tuning (SFT) LLMs to work with retrieval modules (Shi et al., 2025b; Jin et al., 2025; Trivedi et al., 2023; Asai et al., 2023b). These design choices introduce substantial computational overhead, limit the full utilization of the LLM’s shared representational capacity, and increase inference latency due to the need for explicit token-level generation at each step, illustrated in Figure 1.

Recent advances in latent reasoning suggest an alternative paradigm, where LLMs perform reasoning directly in hidden representation space rather than through fully verbalized intermediate steps (Yue et al., 2025; Shi et al., 2025a; Chen et al., 2025d; Xu et al., 2025b). Such approaches demonstrate that latent trajectories in hidden states can encode rich reasoning processes without explicit token generation. However, this paradigm remains largely unexplored in the context of retrieval-augmented systems (Chen et al., 2025d). Extending latent reasoning to RAG is, however, non-trivial. First, unlike standard reasoning tasks, RAG lacks high-quality chain-of-thought supervision for constructing retrieval queries, making it difficult to distill text-based reasoning into latent representations (Hao et al., 2024; Xu et al., 2025a). Second, existing RAG architectures typically rely on separate embedding models for retrieval, which necessitate explicit natural language queries. Consequently, integrating both reasoning traces and retrieval queries into a shared latent space requires a unified model that simultaneously supports retrieval and generation.

To address these challenges, we propose a unified **latent abstraction retrieval-augmented generation** (LAnR) framework, wherein retrieval is conducted directly within the LLM’s internal representation space. Instead of generating textual queries, the model produces latent query vectors derived from the hidden states of designated special tokens

^{*}Equal contribution ¹Center for AI Research, VinUniversity, Vietnam. Correspondence to: Dung D. Le <dung.ld@vinuni.edu.vn>.

Accepted to the ICML 2026 Workshop on Efficient Multimodal Question Answering (EMM-QA), Seoul, South Korea. Copyright 2026 by the author(s).

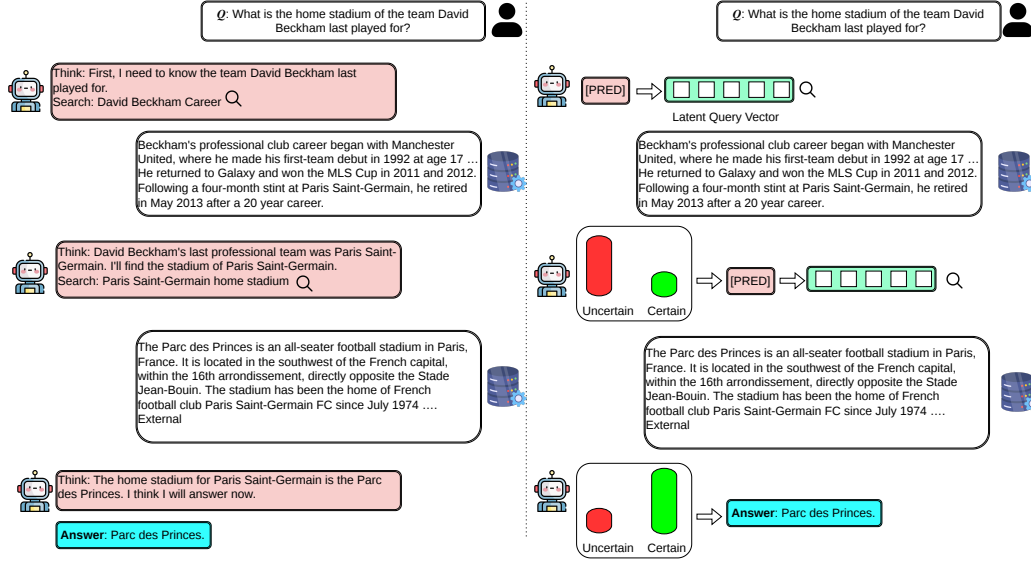


Figure 1. Comparison between conventional RAG and LAnR for multi-hop QA. Conventional RAG performs explicit multi-hop reasoning with intermediate text and queries, whereas LAnR uses latent-space query vectors from [PRED] and an MLP retrieval controller for more efficient integrated reasoning.

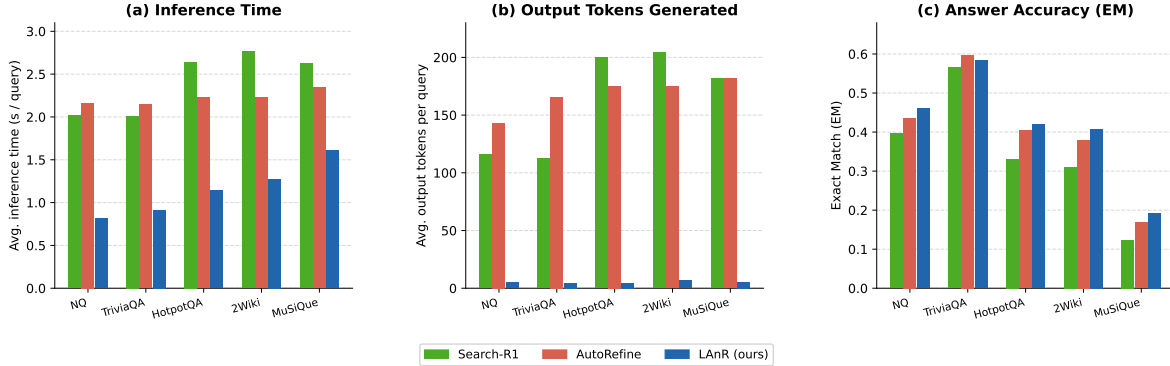


Figure 2. Comparison of inference time, generated tokens, and Exact Match accuracy between prior RAG methods and LAnR. Latent retrieval reduces latency and token generation while maintaining strong performance.

[PRED]. These vectors are used to retrieve documents from a vector database constructed using representations from the same LLM, enabling tight integration between retrieval and generation, detailed in Section 3. To support multi-hop reasoning, we further introduce a lightweight multi-layer perceptron (MLP) (Noriega, 2005) head that predicts whether additional retrieval steps are necessary. This design is motivated by empirical observations that uncertainty signals, such as entropy in the token distribution, correlate with the need for further information acquisition, illustrated in Appendix C. Extensive experiments across multiple benchmarks demonstrate that our approach achieves superior performance compared to state-of-the-art RAG methods, while significantly reducing inference latency due to minimal text generated, illustrated in Figure 2 and eliminating the need for serving a separately trained dense retriever. We provide our code at the anonymous repository <https://anonymous.4open.science/r/LAnR-48EF/>.

[//anonymous.4open.science/r/LAnR-48EF/](https://anonymous.4open.science/r/LAnR-48EF/).

We summarize our main contributions as follows:

1. We propose a novel latent RAG framework that leverages the internal representation space of an LLM to jointly perform document encoding, retrieval, and generation, thereby unifying these components within a single architecture.
2. We design an implicit retrieval control mechanism, implemented as a lightweight auxiliary head, which use the LLM’s hidden representation to adaptively determines the necessity of additional retrieval without relying on explicit text-based reasoning.
3. We introduce a new approach for constructing retrieval query vectors directly from the LLM’s latent representation.

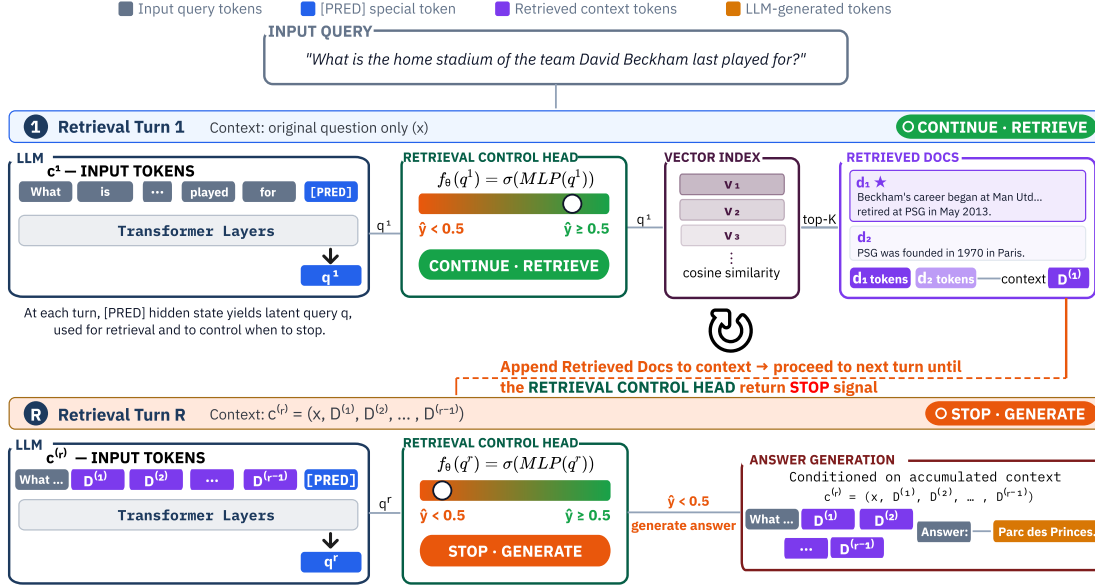


Figure 3. **Overview of LAnR.** Queries are injected into the LLM and combined with a [PRED] token to form a latent query from hidden representations and retrieval control via a lightweight MLP head, after which the LLM generates answers from the retrieved context.

tations, eliminating the need for intermediate natural language query generation.

2. Latent Query Construction

We begin by investigating latent query construction, where the LLM bypasses explicit textual query generation and instead produces a dense retrieval vector directly from its hidden representations. To enable this capability, the LLM is trained to internalize query formulation, allowing it to infer search intent from the input and map it directly into a retrieval vector. Formally, let $x = (x_1, \dots, x_T)$ denote an input query token sequence and $\mathcal{M}$ an autoregressive language model with last-layer hidden states $h_t \in \mathbb{R}^d$. We append a designated token [PRED] to the input, forming $\tilde{x} = (x_1, \dots, x_T, [\text{PRED}])$. Through causal self-attention, its hidden state attends to the full preceding context, yielding a latent query vector:

$$q = h_{[\text{PRED}]} \in \mathbb{R}^d. \quad (1)$$

Optionally, $N \geq 1$ consecutive [PRED] tokens can be appended, allowing the model to construct the query representation over multiple latent steps, which promotes higher-level abstraction and extends the LLM’s latent reasoning process; we default to $N=1$ and study this design choice in Appendix E.

Unlike prior LLM-based encoders (Wang et al., 2024; BehnamGhader et al., 2024), where the query vector encodes a fully-formed textual query, here q is produced by the model’s own reasoning over context. This distinction

becomes essential in the multi-turn setting, detailed in Section 3, where intermediate sub-queries have no canonical textual form and must be inferred from the running context of previously retrieved evidence. Each representation of the document D_i are obtained from the same model $\mathcal{M}$ via last-token pooling (Cheng et al., 2024; Ge et al., 2023):

$$d_i = h_{D_i}^{\text{last}} \in \mathbb{R}^d. \quad (2)$$

Embedding queries and documents within a shared representation space induced by the same parameters eliminates the need for a separate retriever and preserves the model’s generative capability for downstream answer production. We train the [PRED] representation with a standard contrastive objective (Wang et al., 2022; Chen et al., 2024):

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(\text{sim}(q, d^+)/\tau)}{\exp(\text{sim}(q, d^+)/\tau) + \sum_{j=1}^{N^-} \exp(\text{sim}(q, d_j^-)/\tau)} \quad (3)$$

where d^+ is a positive document, $\{d_j^-\}_{j=1}^{N^-}$ are hard negatives mined from the model’s own retrieval errors and periodically refreshed following ANCE (Xiong et al., 2020), $\text{sim}(\cdot, \cdot)$ is cosine similarity, and τ is a temperature parameter. This setup gives us a query vector at any point in the model’s forward pass. Building on this foundation, Section 3 introduces a multi-turn training objective designed to leverage both properties effectively.

3. End-to-End Framework and Training Objective

Building on the latent query construction in Section 2, we extend the framework from single-step implicit retrieval to

a multi-turn setting. The central challenge is to determine *when* to trigger additional retrieval and *what* to retrieve at each step, while relying solely on latent reasoning in hidden space without generating intermediate text tokens. We address both through: (1) an MLP-based retrieval control head that decides whether further retrieval is necessary, and (2) an adaptive contrastive target mechanism that dynamically updates the retrieval objective based on remaining unretrieved evidence. We illustrate the overview of our method in Figure 3.

At each retrieval turn $r = 1, 2, \dots, R$, the model constructs a latent query from the current context, which now includes both the original question and any previously retrieved documents, and decides whether to retrieve again or to proceed with answer generation. Let $c^{(r)}$ denote the accumulated context at turn r :

$$c^{(r)} = (x, D^{(1)}, D^{(2)}, \dots, D^{(r-1)}),$$

where $D^{(s)}$ denotes the set of top- K documents retrieved at turn s . At each turn, we append [PRED] to $c^{(r)}$ and extract the latent query vector: $q^{(r)} = h_{[\text{PRED}]}^{(r)} \in \mathbb{R}^d$, which is used both for retrieval via similarity matching against the document index, detailed in Equation 2 and as input to the retrieval control head described below. The process repeats until the control head signals termination or a maximum number of turns R is reached, after which the model generates the final answer conditioned on the full accumulated context $c^{(R+1)}$.

MLP-based Retrieval Control Head. An additional component of our framework is a lightweight MLP head that determines whether the currently retrieved evidence is sufficient to answer the query, or whether additional retrieval is required. At each turn r , the Retrieval Control Head f_θ takes $q^{(r)}$ as input and produces a binary decision:

$$\hat{y}^{(r)} = f_\theta(q^{(r)}) = \sigma(\text{MLP}(q^{(r)})) \in [0, 1], \quad (4)$$

where σ is the sigmoid function. A prediction $\hat{y}^{(r)} < 0.5$ indicates that all necessary evidence has been retrieved and the model should proceed to generation; $\hat{y}^{(r)} \geq 0.5$ signals that additional retrieval is needed. The training label is derived from the model’s own retrieval state. Let $\mathcal{P}$ denote the full set of positive (gold) documents required to answer the query, and $\mathcal{R}^{(r)} = D^{(1)} \cup \dots \cup D^{(r)}$ the documents retrieved up to turn r . The ground-truth label is:

$$y^{(r)} = \begin{cases} 0, & \text{if } \mathcal{P} \subseteq \mathcal{R}^{(r)}, \\ 1, & \text{otherwise,} \end{cases} \quad (5)$$

i.e., the label is 0 (stop) when all positive documents have been successfully retrieved, and 1 (continue) otherwise. This self-supervised formulation requires no external oracle:

the signal is generated entirely from the model’s own retrieval accuracy at training time. The control head is trained with binary cross-entropy:

$$\mathcal{L}_{\text{ctrl}} = - \sum_{r=1}^R \left[y^{(r)} \log \hat{y}^{(r)} + (1 - y^{(r)}) \log (1 - \hat{y}^{(r)}) \right]. \quad (6)$$

In the multi-turn setting, naively using the same contrastive target at every turn is suboptimal: once a positive document has been retrieved, it should no longer serve as the target for subsequent queries. At turn r , the set of leftover positive documents is: $\mathcal{P}^{(r)} = \mathcal{P} \setminus \mathcal{R}^{(r-1)}$, where $\mathcal{R}^{(0)} = \emptyset$. The contrastive loss at turn r uses a positive document sampled from $\mathcal{P}^{(r)}$:

$$\mathcal{L}_{\text{CL}}^{(r)} = - \log \frac{\exp(\text{sim}(q^{(r)}, d^{+(r)})/\tau)}{\exp(\text{sim}(q^{(r)}, d^{+(r)})/\tau) + \sum_{j=1}^{N^-} \exp(\text{sim}(q^{(r)}, d_j^-)/\tau)}, \quad (7)$$

where $d^{+(r)}$ is the embedding of a document drawn from $\mathcal{P}^{(r)}$. This steers each successive turn toward the missing evidence, avoiding redundant retrieval. The full training loss combines next-token prediction, multi-turn contrastive retrieval, and the control head objective:

$$\mathcal{L} = \sum_{r=1}^R \mathcal{L}_{\text{CL}}^{(r)} + \lambda \mathcal{L}_{\text{NTP}} + \mu \mathcal{L}_{\text{ctrl}}, \quad (8)$$

where λ and μ are hyperparameters balancing the three objectives. $\mathcal{L}_{\text{NTP}}$ denotes the standard next-token prediction loss, which maintains the generative capability of the LLM. We apply loss masking to retrieved tokens and [PRED] tokens, so that the optimization objective is computed only over tokens generated by the LLM, excluding retrieved content from gradient updates, following prior works (Jin et al., 2025; Shi et al., 2025b).

Inference. During inference, the model performs iterative retrieval beginning from the input query. At each step, the model appends the [PRED] token, extracts the latent query representation $q^{(r)}$, and feeds it into the retrieval control head. If the controller predicts that additional retrieval is needed ($\hat{y}^{(r)} \geq 0.5$), the model retrieves the top- K documents, appends them to the current context, and continues the retrieval process. Otherwise, the model stops retrieving and generates the final answer conditioned on the accumulated evidence. To prevent unbounded retrieval, we impose a maximum limit of $R = 4$ retrieval rounds. Additional qualitative examples are provided in Appendix F.

4. Experiments

In this section, we investigate the following research questions (RQs):

RQ1. How effective is LAnR compared to conventional agentic RAG systems in answer quality across single-hop

and multi-hop benchmarks, and can its implicit retriever match text embedding models in retrieval accuracy?

RQ2. Does LAnR’s retrieval control head adaptively allocate search hops based on query complexity, and does each retrieval hop find relevant evidence more effectively than text-based iterative search?

RQ3. Can LAnR reduce inference time by minimizing retrieval iterations and improving efficiency through more abstract query representations?

RQ4. What is the individual contribution of each training objective ($\mathcal{L}_{\text{NTP}}$, $\mathcal{L}_{\text{CL}}$, $\mathcal{L}_{\text{ctrl}}$) and the retrieval control head to LAnR’s end-to-end performance?

4.1. Experimental Setup

Settings. We evaluate our method on five open-domain QA benchmarks, including two single-hop datasets, Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), and three multi-hop datasets: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (2Wiki) (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We report Exact Match (EM) as the primary metric for answer generation. For retrieval evaluation on datasets with ground-truth documents, we use Recall as the main metric.

We compare LAnR against three categories of methods: (1) generation without retrieval, including direct LLM generation and supervised fine-tuning (SFT) without retrieval; (2) single-hop retrieval methods, such as naive RAG (Lewis et al., 2020) that retrieves once using the input query; and (3) multi-hop retrieval approaches, including agentic and iterative systems such as Search-o1 (Li et al., 2025b), IR-CoT (Trivedi et al., 2023), Search-R1 (Jin et al., 2025), and AutoRefine (Shi et al., 2025b).

Implementation Details. To simulate a realistic retrieval setting, we remove ground-truth context from the QA datasets and instead use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the external knowledge source. By default, retrieval returns the top- k documents at each step, with $k = 3$ following the setting of Search-R1 for fair comparison. All models are trained on the combined NQ and HotpotQA dataset, following the setup of prior works (Jin et al., 2025; Shi et al., 2025b). Detailed training configurations are described in Appendix B.

4.2. Main Results

Overall Performance (RQ1). Table 1 reports EM accuracy across five QA benchmarks covering both single-hop and multi-hop reasoning. Retrieval-free methods perform poorly, while Naive RAG improves single-hop QA but remains ineffective on compositional tasks such as HotpotQA, 2Wiki, and MuSiQue. Among multi-hop ap-

proaches, Search-R1 and AutoRefine provide strong baselines, but LAnR consistently achieves the best overall performance, especially on challenging multi-hop benchmarks. LAnR-Instruct reaches 0.419 EM on HotpotQA, 0.408 on 2Wiki, and 0.193 on MuSiQue, outperforming AutoRefine-Instruct across all three datasets. Although AutoRefine remains competitive on single-hop benchmarks, LAnR achieves the highest average EM overall, indicating a stronger balance between retrieval and reasoning. Despite being trained only on NQ and HotpotQA, LAnR also generalizes effectively to unseen datasets in a zero-shot setting.

Retrieval Quality vs. Text Embedding Retrievers. Table 2 compares LAnR-Instruct with sparse and dense retrievers across three multi-hop QA benchmarks. While BM25, BGE, and E5 rely on fixed top- K retrieval and are trained solely for retrieval, LAnR-Instruct jointly performs adaptive retrieval and generation. Despite this joint objective, LAnR-Instruct achieves competitive retrieval recall using substantially fewer retrieved documents. On HotpotQA and 2WikiMQA, it attains recalls of 0.840 and 0.715 with only ~ 5 retrieved documents, matching or exceeding the Recall@5-10 performance of specialized retrievers. On the more challenging MuSiQue benchmark, LAnR-Instruct achieves 0.516 recall with ~ 7.9 documents, outperforming BM25 and remaining competitive with dense retrievers. In contrast, the single-step LAnR-Instruct-static variant consistently underperforms the adaptive version, highlighting the benefit of iterative retrieval with learned control.

Latent RAG Effectiveness (RQ2). To understand *how* LAnR retrieves rather than just *how well*, we analyse the search-level behaviour of the retrieval control head across all five evaluation benchmarks.

Adaptive search count. Figure 4(a) shows the distribution of search calls per query across all five datasets. On single-hop benchmarks (NQ, TriviaQA), the vast majority of queries are resolved in a single search call, reflecting the control head’s ability to recognise when sufficient evidence has been gathered early. On compositional multi-hop benchmarks the distribution shifts substantially: 2Wiki and MuSiQue queries spread across two and three search calls, with MuSiQue requiring the most, consistent with its longer evidence chains.

Step-wise retrieval quality. Figure 4(b) compares cumulative gold recall across retrieval hops between LAnR and AutoRefine on the three multi-hop benchmarks. LAnR consistently achieves higher recall at every step, with the advantage appearing from the first hop and widening on more compositional datasets such as 2Wiki and MuSiQue. This suggests that latent query vectors better capture residual information needs, enabling subsequent retrieval steps to target missing evidence more effectively than text-based queries.

Table 1. (RQ1) Accuracy comparison of LAnR and baselines on QA benchmarks. **Bold** and underline denote best and second-best results, respectively. "-Instruct" and "-Base" indicate the corresponding Qwen2.5-3B backbone variants.

Methods	Single-Hop QA		Multi-Hop QA			
	NQ	TriviaQA	HotpotQA	2Wiki	Musique	Avg.
<i>w/o Retrieval</i>						
Direct Generation	0.106	0.288	0.149	0.244	0.020	0.134
SFT	0.249	0.292	0.186	0.248	0.044	0.176
<i>w/ Single-Hop Retrieval</i>						
Naive RAG (Lewis et al., 2020)	0.348	0.544	0.255	0.226	0.047	0.270
<i>w/ Multi-Hop Retrieval</i>						
Search-ol (Li et al., 2025b)	0.238	0.472	0.221	0.218	0.054	0.255
IRCoT (Trivedi et al., 2023)	0.111	0.312	0.164	0.171	0.067	0.181
ReSearch-Instruct (Chen et al., 2025b)	0.365	0.571	0.351	0.272	0.095	0.331
ReSearch-Base (Chen et al., 2025b)	0.427	0.597	0.305	0.272	0.074	0.319
Search-R1-Instruct (Jin et al., 2025)	0.397	0.565	0.331	0.310	0.124	0.336
Search-R1-Base (Jin et al., 2025)	0.421	0.583	0.297	0.274	0.066	0.312
AutoRefine-Instruct (Shi et al., 2025b)	0.436	0.597	0.404	0.380	0.169	0.396
AutoRefine-Base (Shi et al., 2025b)	0.467	0.620	0.405	0.393	0.157	0.405
LAnR-Instruct	<u>0.460</u>	<u>0.613</u>	0.419	0.408	0.193	0.418
LAnR-Base	0.455	0.610	<u>0.417</u>	<u>0.402</u>	<u>0.187</u>	<u>0.414</u>

Table 2. Retrieval recall at varying document budgets. BM25, BGE, and E5 are trained with retrieval supervision only. LAnR-Instruct-static employs the single-step retrieved [PRED] embedding as a fixed query vector, following the procedure described in Section 2; LAnR-Instruct uses adaptive multi-hop retrieval with a learned controller. Despite joint training with generation, LAnR-Instruct matches or exceeds specialized retrievers at comparable document budgets.

Dataset	Method	R@1	R@3	R@5	R@10	Recall (budget)
HotpotQA	BM25 (Robertson & Zaragoza, 2009)	0.401	0.603	0.670	0.751	—
	BGE (Chen et al., 2024)	0.477	0.782	0.832	0.880	—
	E5 (Wang et al., 2022)	0.462	0.773	0.826	0.875	—
	LAnR-Instruct-static	0.451	0.769	0.800	0.830	—
	LAnR-Instruct	—	—	—	—	0.840±0.0012 (~5.5 docs)
2WikiMQA	BM25 (Robertson & Zaragoza, 2009)	0.360	0.555	0.605	0.656	—
	BGE (Chen et al., 2024)	0.411	0.643	0.680	0.715	—
	E5 (Wang et al., 2022)	0.415	0.656	0.687	0.717	—
	LAnR-Instruct-static	0.407	0.641	0.684	0.713	—
	LAnR-Instruct	—	—	—	—	0.715±0.0009 (~5.1 docs)
MuSiQue	BM25 (Robertson & Zaragoza, 2009)	0.216	0.310	0.350	0.400	—
	BGE (Chen et al., 2024)	0.280	0.435	0.500	0.573	—
	E5 (Wang et al., 2022)	0.290	0.417	0.454	0.503	—
	LAnR-Instruct-static	0.265	0.393	0.449	0.487	—
	LAnR-Instruct	—	—	—	—	0.516±0.0011 (~7.9 docs)

Control head precision. Figure 4(c) reports the productive hop rate: the fraction of additional search calls triggered by a "continue" decision that successfully retrieved at least one new gold document. On 2Wiki, 75% of additional calls are productive, confirming that the control head issues follow-up searches only when evidence is genuinely incomplete. On MuSiQue, the rate is lower at 23%, reflecting the difficulty of locating the precise gold documents in highly compositional three-hop and four-hop chains where missing any single query may fail to bridge the full reasoning gap.

Inference Efficiency of Latent Retrieval (RQ3). Figure 5 directly tests the core efficiency claim of LAnR: each subplot shows the EM distribution for all three methods on a single benchmark, with the colored x-tick labels reporting each method’s average number of retrieval calls.

Across all five datasets, LAnR issues the fewest retrieval calls (1.54 – 2.47), yet its median EM matches or exceeds AutoRefine (2.02 – 2.13 calls) and Search-R1 (2.01 – 3.13 calls). The narrow interquartile ranges confirm that LAnR’s accuracy is stable across evaluation instances rather than driven by a subset of easy queries. Notably, on the multi-hop benchmarks, LAnR’s gap over Search-R1 widens substantially despite Search-R1 issuing nearly twice as many retrievals, indicating that latent query vectors retrieve more relevant evidence per call than token-based queries.

The token and latency costs are quantified in Figure 2. LAnR generates on average only ~5 output tokens per query, roughly 30× fewer than Search-R1 (~163) or AutoRefine (~168), because the retrieval signal is encoded as a latent vector rather than verbalized text. This reduction translates directly to wall-clock savings: LAnR completes a query in

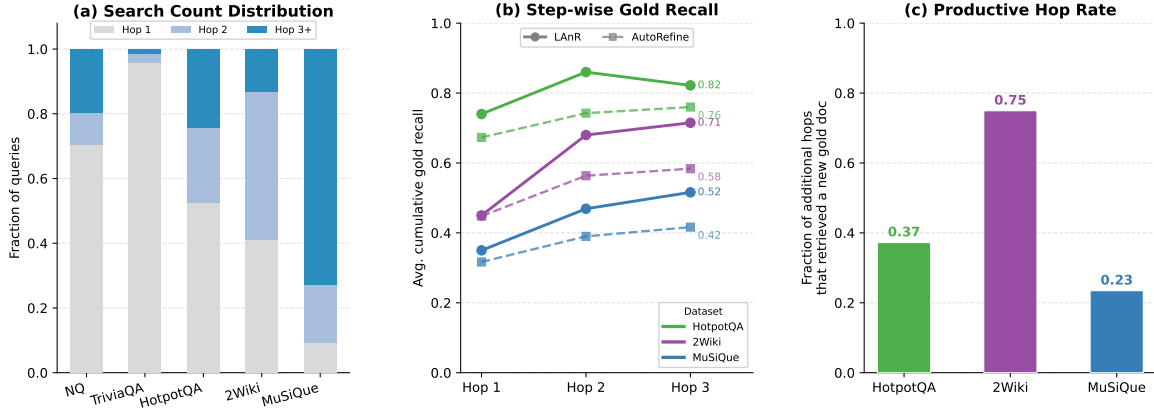


Figure 4. (RQ2) LAnR adaptive search behaviour. (a) Distribution of search counts per query across all five benchmarks. Single-hop datasets are dominated by a single search call, while compositional benchmarks require progressively more, showing the control head adapts to query complexity rather than applying a fixed budget. (b) Step-wise cumulative gold recall at each hop for LAnR (solid) and AutoRefine (dashed) on HotpotQA, 2Wiki, and MuSiQue. LAnR consistently outperforms AutoRefine at every hop on all three datasets. (c) Productive hop rate: fraction of additional search calls ($\geq$ Hop 2) that retrieved at least one new gold document per dataset.

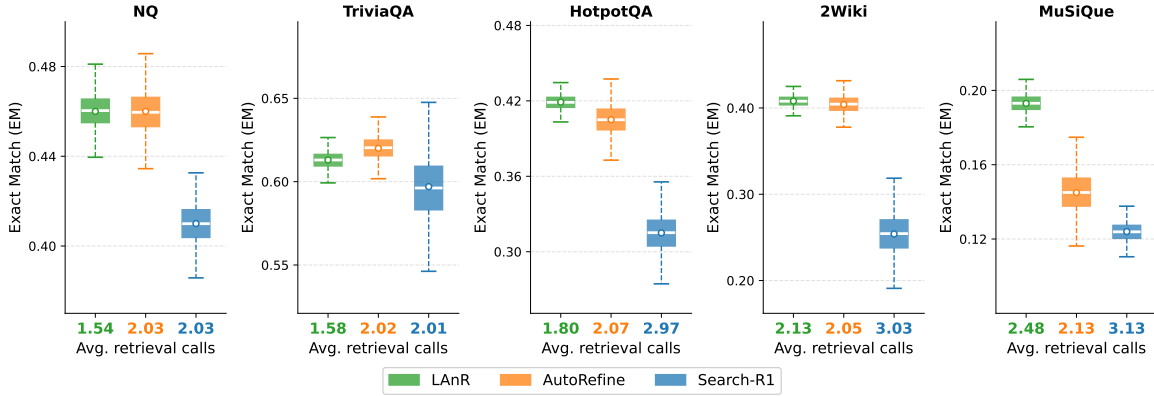


Figure 5. (RQ3) Per-dataset EM distributions for LAnR, AutoRefine, and Search-R1. LAnR achieves competitive or higher EM with the fewest retrieval calls and consistently narrow variance.

0.81 – 1.61 s compared to 2.01 – 2.77 s for Search-R1 and 2.15 – 2.34 s for AutoRefine, a $1.5 - 2.7\times$ speedup across all benchmarks.

Key Takeaway. By replacing explicit query generation with compact latent vectors and an adaptive retrieval controller, LAnR performs fewer but more effective search calls, achieves higher cumulative gold recall, and maintains stable EM performance with lower computational cost.

4.3. Ablation Studies

Component ablation (RQ4). Table 3 isolates the contribution of three design choices: the MLP retrieval control head, NTP loss masking, and the NTP loss weight λ_{NTP} . Removing the control head and replacing it with a fixed 5-document budget leaves single-hop performance essen-

tially unchanged but causes large drops on multi-hop tasks, confirming that adaptive retrieval is the primary driver of multi-hop gains. Removing NTP loss masking introduces a moderate but consistent degradation on all three multi-hop benchmarks, suggesting the masking prevents the model from conflating retrieval-query generation with next-token prediction over retrieved content. Reducing λ_{NTP} to 0.5 incurs a small performance drop, while setting it to 0 collapses performance to near zero, demonstrating that the NTP objective is indispensable for answer generation.

Additional Experimental Results. Figure 6 shows EM as a function of training-set fraction for both LAnR-3B and LAnR-7B across all five benchmarks. The learning curve rises steeply in the first 10-20% of data, where the four-dataset average EM climbs from 0.13 to 0.38. Notably, 2WikiMultihopQA starts near zero at 5% and gains most rapidly between 10-50%, reflecting the higher sample

Table 3. Component ablation study (RQ4). Δ Avg denotes the absolute decrease in average performance relative to the full LAnR model.

Variant	NQ	TriviaQA	HotpotQA	2Wiki	MuSiQue	Avg	Δ Avg
LAnR-Instruct	0.460	0.613	0.419	0.408	0.193	0.418	—
w/o ctrl head	0.459	0.592	0.342	0.266	0.095	0.350	−0.068
w/o NTP loss masking	0.458	0.582	0.395	0.371	0.182	0.398	−0.020
$\lambda_{\text{NTP}} = 0.5$	0.455	0.578	0.411	0.402	0.189	0.407	−0.011
$\lambda_{\text{NTP}} = 0$	0.005	0.007	0.004	0.000	0.000	0.003	−0.415

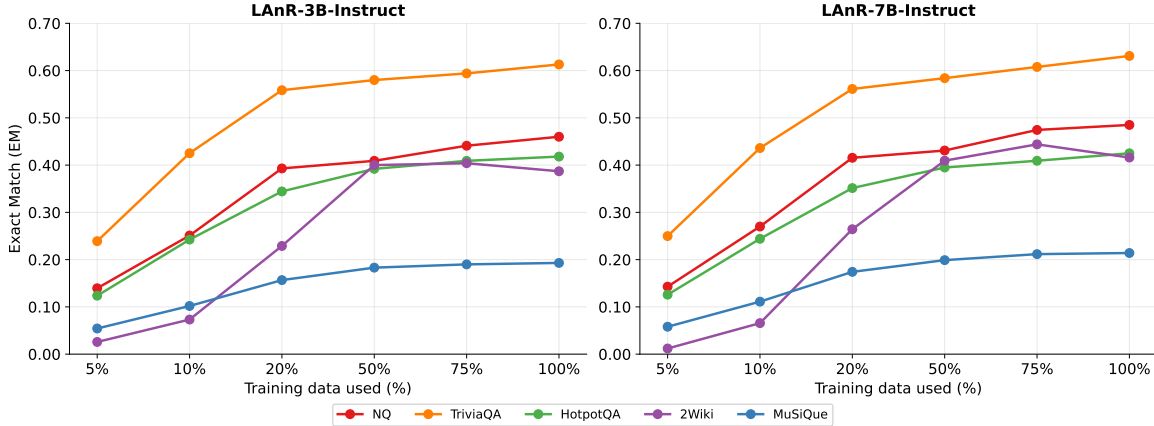


Figure 6. EM at various fractions of training data used (5 % to 100 %). Most of the performance gain is captured by the first 50 % of the data.

complexity of multi-hop compositional reasoning compared to single-hop benchmarks (NQ, TriviaQA), which plateau earlier. Beyond 50% gains flatten to within 3-5% of the full-data ceiling, suggesting that LAnR’s latent retrieval mechanism generalises effectively with moderate training budgets. The 7B variant yields a consistent improvement over 3B across all data fractions, indicating that model scale and training data contribute largely independently to final performance.

Additional analyses are provided to further examine LAnR from multiple perspectives, including: (i) a detailed evaluation of the Retrieval Control Head, Appendix E.1; (ii) model scaling experiments spanning 3B to 7B backbones, Appendix E.2; (iii) statistical significance analysis across multiple random seeds, Appendix E.3; and (iv) the effect of varying the number of [PRED] tokens, where larger token counts improve abstraction capacity, Appendix E.4. Together, these findings further demonstrate the robustness, scalability, and retrieval effectiveness of LAnR across diverse QA benchmarks.

5. Related Works

Recent advances in latent reasoning (Yue et al., 2025; Hao et al., 2024; Xu et al., 2025a; Yu et al., 2026) explore performing reasoning directly in continuous hidden-state space rather than through explicit chain-of-thought decoding, en-

abling richer representations and more efficient inference. Meanwhile, retrieval-augmented generation (RAG) methods improve factuality through external retrieval (Asai et al., 2023b; Trivedi et al., 2023; Karpukhin et al., 2020), with recent work focusing on adaptive and iterative retrieval strategies (Jin et al., 2025; Li et al., 2025b). However, existing approaches largely rely on discrete text-based reasoning and retrieval control. We provide the full related works discussion in Appendix A. We provide the full related works section in Appendix A.

6. Conclusion

We introduced **latent abstraction retrieval-augmented generation** (LAnR), a unified framework that performs retrieval and reasoning directly in the latent space of a single LLM. By replacing explicit text-based query generation with latent query vectors derived from hidden states, LAnR eliminates the need for a separate retriever and reduces reliance on token-level reasoning. We further proposed a lightweight retrieval control head that adaptively determines when additional retrieval is required, enabling efficient multi-hop reasoning without explicit intermediate text. Empirical results across multiple benchmarks demonstrate that LAnR achieves competitive performance compared to existing RAG systems while significantly reducing inference latency. These findings highlight the potential of latent-

space reasoning as a scalable alternative to conventional RAG pipelines.

Impact Statement

LAnR is foundational research on latent retrieval-augmented generation efficiency and does not target any specific deployment. Nevertheless, we briefly discuss potential societal implications.

Positive impacts. By reducing inference latency by up to $2.7\times$ and token generation by $\approx 30\times$ compared to existing RAG systems, LAnR lowers the computational cost of knowledge-grounded language model inference, improving accessibility for resource-constrained practitioners and reducing energy consumption at scale. Improved factuality in question answering also has broad benefits in educational, medical, and scientific information access.

Potential negative impacts. As with any system that improves the fluency and factual grounding of LLM outputs, LAnR could lower the barrier to generating more convincing disinformation or fabricated content at scale. The efficiency gains in particular may make high-volume automated generation more feasible. Additionally, because LAnR retrieves from large external corpora, any biases present in those corpora may be silently propagated into generated answers without explicit query traces, potentially making such biases harder to audit than in text-based retrieval pipelines where queries are inspectable.

References

- Arslan, M., Ghanem, H., Munawar, S., and Cruz, C. A survey on rag with llms. *Procedia computer science*, 246: 3781–3790, 2024.
- Asai, A., Min, S., Zhong, Z., and Chen, D. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pp. 41–46, 2023a.
- Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023b.
- BehnamGhader, P., Adlakha, V., Mosbach, M., Bahdanau, D., Chapados, N., and Reddy, S. Llm2vec: Large language models are secretly powerful text encoders, 2024. URL <https://arxiv.org/abs/2404.05961>, 2024.
- Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.
- Chen, M., Sun, L., Li, T., Sun, H., Zhou, Y., Zhu, C., Wang, H., Pan, J. Z., Zhang, W., Chen, H., et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025a.
- Chen, M., Sun, L., Li, T., Sun, H., Zhou, Y., Zhu, C., Wang, H., Pan, J. Z., Zhang, W., Chen, H., et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025b.
- Chen, X., Zhao, A., Xia, H., Lu, X., Wang, H., Chen, Y., Zhang, W., Wang, J., Li, W., and Shen, X. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025c.
- Chen, X., Zhao, A., Xia, H., Lu, X., Wang, H., Chen, Y., Zhang, W., Wang, J., Li, W., and Shen, X. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025d.
- Cheng, X., Wang, X., Zhang, X., Ge, T., Chen, S.-Q., Wei, F., Zhang, H., and Zhao, D. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.
- Das, D., O’Nuallain, S., and Rahimi, R. Rader: Reasoning-aware dense retrieval models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19981–20008, 2025.
- Deng, J., Jiang, Z., Pang, L., Wei, Z., Chen, L., Xu, K., Song, Y., Shen, H., and Cheng, X. Following the autoregressive nature of llm embeddings via compression and alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12672–12688, 2025.
- Ge, T., Hu, J., Wang, L., Wang, X., Chen, S.-Q., and Wei, F. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.
- Goyal, S., Ji, Z., Rawat, A. S., Menon, A. K., Kumar, S., and Nagarajan, V. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.
- Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

- Ho, X., Nguyen, A.-K. D., Sugawara, S., and Aizawa, A. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.
- Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.
- Jin, B., Yoon, J., Han, J., and Arik, S. O. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*, 2024.
- Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., and Han, J. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.
- Kim, C. and Kim, S. Freeson: Retriever-free retrieval-augmented reasoning via corpus-traversing mcts. *arXiv preprint arXiv:2505.16409*, 2025.
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.
- Li, J., Fu, Y., Fan, L., Liu, J., Shu, Y., Qin, C., Yang, M., King, I., and Ying, R. Implicit reasoning in large language models: A comprehensive survey. *arXiv preprint arXiv:2509.02350*, 2025a.
- Li, X., Dong, G., Jin, J., Zhang, Y., Zhou, Y., Zhu, Y., Zhang, P., and Dou, Z. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025b.
- Nie, Z., Feng, Z., Li, M., Zhang, C., Zhang, Y., Long, D., and Zhang, R. When text embedding meets large language model: a comprehensive survey. *arXiv preprint arXiv:2412.09165*, 2024.
- Noriega, L. Multilayer perceptron tutorial. *School of Computing. Staffordshire University*, 4(5):444, 2005.
- Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.
- Robertson, S. and Zaragoza, H. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.
- Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.
- Shi, D., Asi, A., Li, K., Yuan, X., Pan, L., Lee, W., and Xiao, W. Swireasoning: Switch-thinking in latent and explicit for pareto-superior reasoning llms. *arXiv preprint arXiv:2510.05069*, 2025a.
- Shi, Y., Li, S., Wu, C., Liu, Z., Fang, J., Cai, H., Zhang, A., and Wang, X. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *arXiv preprint arXiv:2505.11277*, 2025b.
- Silva, L. and Barbosa, L. Improving dense retrieval models with llm augmented data for dataset search. *Knowledge-based systems*, 294:111740, 2024.
- Springer, J. M., Kotha, S., Fried, D., Neubig, G., and Raghunathan, A. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*, 2024.
- Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.
- Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

- Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
- Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, 2024.
- Wang, X., Wang, S., Zhu, Y., and Liu, B. System-1.5 reasoning: Traversal in language and latent spaces with dynamic shortcuts. *arXiv preprint arXiv:2505.18962*, 2025.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., Ahmed, J., and Overwijk, A. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.
- Xu, Y., Guo, X., Zeng, Z., and Miao, C. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351, 2025a.
- Xu, Y., Guo, X., Zeng, Z., and Miao, C. Softcot++: Test-time scaling with soft chain-of-thought reasoning. *arXiv preprint arXiv:2505.11484*, 2025b.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.
- Yu, X., Chen, Z., He, Y., Fu, T., Yang, C., Xu, C., Ma, Y., Hu, X., Cao, Z., Xu, J., et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026.
- Yue, Z., Zhuang, H., Bai, A., Hui, K., Jagerman, R., Zeng, H., Qin, Z., Wang, D., Wang, X., and Bendersky, M. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024.
- Yue, Z., Jin, B., Zeng, H., Zhuang, H., Qin, Z., Yoon, J., Shang, L., Han, J., and Wang, D. Hybrid latent reasoning via reinforcement learning. *arXiv preprint arXiv:2505.18454*, 2025.

A. Full Related Works

A.1. Latent Reasoning

A key limitation of dominant reasoning paradigms such as explicit chain-of-thought (CoT) (Wei et al., 2022; Yao et al., 2023; Goyal et al., 2023; Yu et al., 2026) lies in their reliance on discrete token generation during inference. In standard CoT decoding, the model commits to a single token at each step, sampled from the predicted distribution. While this process enhances interpretability by verbalizing intermediate reasoning steps, it inherently collapses the full probability distribution into a single trajectory, discarding uncertainty and eliminating alternative reasoning paths that may be informative. To address this limitation, recent work has explored latent reasoning (Hao et al., 2024; Xu et al., 2025a; Shi et al., 2025a; Yue et al., 2025; Wang et al., 2025), where reasoning unfolds directly in the continuous hidden-state space rather than through discrete text. This paradigm offers two primary advantages over CoT: (1) increased representational capacity per step, as continuous vectors can encode substantially richer information than individual tokens (Chen et al., 2025c); and (2) the ability to implicitly preserve multiple reasoning hypotheses without prematurely collapsing them into a single token sequence (Li et al., 2025a). However, despite these advances, prior work on latent reasoning has primarily focused on improving reasoning quality in standalone language modeling settings. Its application to RAG remains largely unexplored. In particular, existing methods do not address how latent representations can be leveraged for retrieval itself, nor how they can guide adaptive retrieval decisions.

A.2. Retrieval Augmented Generation

RAG enhances LLMs by incorporating external knowledge sources to mitigate hallucinations and address knowledge gaps (Lewis et al., 2020; Yue et al., 2024; Arslan et al., 2024). A central challenge in RAG systems lies in determining when and how to retrieve relevant information, as naive single-step retrieval often introduces irrelevant or insufficient context (Jiang et al., 2023; Jin et al., 2024). Early approaches employ supervised fine-tuning (SFT) to train models for query generation and retrieval integration (Asai et al., 2023b; Trivedi et al., 2023; Yao et al., 2022; Schick et al., 2023); however, these methods rely on high-quality annotated trajectories and often exhibit limited generalization to out-of-distribution settings. Leveraging the inherent structure of LLMs, an alternative direction is to fine-tune them as document encoders for text embedding (BehnamGhader et al., 2024; Wang et al., 2024; Springer et al., 2024; Deng et al., 2025), or to use LLMs to generate synthetic data to support embedding training (Das et al., 2025; Nie et al., 2024; Silva & Barbosa, 2024). More recent work explores iterative and adaptive retrieval strategies, where models interleave reasoning and retrieval in a multi-step process, commonly described as "search-during-think" (Jin et al., 2025; Shi et al., 2025b; Chen et al., 2025a; Kim & Kim, 2025). However, existing approaches primarily operate in the discrete text space, relying on explicit query generation and token-level signals to control retrieval. Moreover, they often lack mechanisms for directly assessing retrieval sufficiency or refining retrieved evidence beyond surface-level interactions.

B. More Implementation Details

Table 4. Primary hyperparameters used by LAnR.

Hyper-parameter	Value
Training Batch Size	256
Chunk size	100 words
Micro Training Batch Size	32
Learning Rate	5×10^{-5}
Validation Batch Size	256
Total Training Steps	250
Max Search Actions	4
λ for NTP Loss	1
μ for Retrieval Control Head Loss	1

Training Details We train LAnR using full-parameter fine-tuning on 2 NVIDIA H100 (80GB) GPUs. The training data is constructed by combining Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018), following a consistent setup across LAnR and all training-based baselines. We employ Fully Sharded Data Parallelism

Table 5. Statistics of the datasets used in this paper.

	NQ	TriviaQA	HotpotQA	2Wiki	Musique
Train	79168	78785	90447	15,000	19,938
Dev	8757	8837	7405	12576	2417
Test	3610	11313	-	-	-

 Table 6. **Serving efficiency comparison.** Both Auto-Refine-Instruct and LAnR-Instruct use the same Qwen2.5-3B backbone. Auto-Refine-Instruct requires an additional external embedding model for retrieval, whereas LAnR-Instruct performs retrieval and generation within a unified model.

Method	Retrieval Architecture	Extra Retriever	VRAM Usage (GB BF16)
Auto-Refine-Instruct	Text-based retrieval	Yes	6.48
LAnR-Instruct	Unified latent retrieval	No	5.85

(FSDP) for distributed training and use bfloat16 precision for both training and evaluation. Table 4 summarizes the main hyperparameters. During inference, we sample with a temperature of 1.0 and allow up to 4 retrieval steps per query. Retrieved documents are concatenated and truncated to a maximum length of 512 tokens. For direct inference and supervised fine-tuning (SFT) baselines, we adopt Qwen2.5-3B-Instruct (Yang et al., 2025) as the backbone model.

Dataset Statistics. All datasets are obtained from the FlashRAG Datasets collection¹. Detailed statistics are reported in Table 5. The LAnR training set is constructed from the training splits of NQ and HotpotQA, comprising 169,615 examples. For evaluation, we aggregate the test or development splits from seven benchmarks: test splits are used for datasets that provide them (e.g., NQ, TriviaQA), while development splits are used otherwise (e.g., HotpotQA, 2Wiki, MuSiQue).

Computational serving. LAnR-Instruct reduces serving-time computational overhead by unifying retrieval and generation within a single model. Table 6 compares LAnR-Instruct with Auto-Refine-Instruct using the same Qwen2.5-3B backbone. Unlike Auto-Refine-Instruct, which requires an additional external embedding retriever during inference, LAnR-Instruct performs retrieval directly through latent representations inside the generation model. As a result, LAnR-Instruct requires lower VRAM usage and simplifies deployment by eliminating the need to maintain a separate text retrieval encoder.

B.1. Retrieval Control Head Model Design

The retrieval control head f_θ is a single affine projection followed by a sigmoid:

$$\hat{y}^{(r)} = \sigma(W q^{(r)} + b), \quad W \in \mathbb{R}^{1 \times d}, \quad b \in \mathbb{R}, \quad (9)$$

where $q^{(r)} = h_{[\text{PRED}]}^{(r)} \in \mathbb{R}^d$ is the last-layer hidden state of the final [PRED] token at retrieval turn r , and d is the LLM’s hidden dimension ($d=2048$ for Qwen-2.5-3B, $d=3584$ for Qwen-2.5-7B). The head introduces $d+1$ trainable parameters, 2,049 for the 3B backbone and 3,585 for the 7B backbone, adding negligible capacity (under 0.0002% of total model parameters) and requiring zero additional LLM forward passes at inference.

C. Latent Retrieval Control

Existing iterative RAG systems rely on explicit token-level signals to determine retrieval termination (Trivedi et al., 2023; Li et al., 2025b; Jin et al., 2025), entangling the stopping decision with surface-level text generation rather than the model’s internal assessment of evidence sufficiency. This raises a natural question: textit Do the hidden states of an LLM already encode a reliable signal of whether the retrieved context is adequate, even before any answer tokens are produced?

Setup. Given an input query token sequence x with an instruction to produce an answer, and a subset $\mathcal{G}_k \subseteq \mathcal{P}$ of k gold supporting documents sampled from the full set $\mathcal{P}$, we construct a probe sequence that enforces answer generation conditioned on the provided context:

$$\tilde{x}_k = (x, \mathcal{G}_k, [\text{PRED}]), \quad (10)$$

¹https://huggingface.co/datasets/RUC-NLP/FlashRAG_datasets

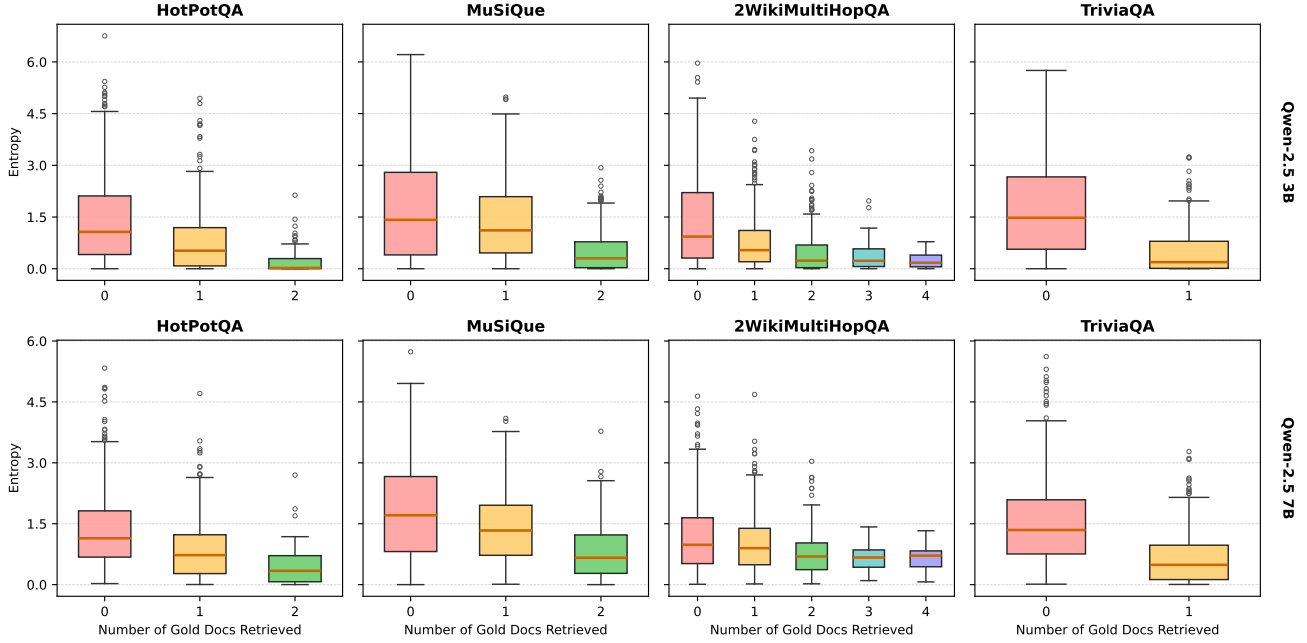


Figure 7. Entropy distribution on RAG benchmarks using Qwen models when answers are generated with gold documents provided in context. Increasing the number of gold documents leads to lower entropy, indicating more confident predictions.

By causal attention, the hidden state $h_k := h_{[\text{PRED}]}(\tilde{x}_k) \in \mathbb{R}^d$ attends to the full query and gold context. The LM head induces a distribution over the first answer token, $p_k(v) = \text{softmax}(Wh_k)_v$, with predictive entropy: $H_k = -\sum_{v \in \mathcal{V}} p_k(v) \log p_k(v)$, where $\mathcal{V}$ denotes the vocabulary.

Experimental Findings. We vary k from 0, as no gold evidence to $k_{\max} = |\mathcal{P}|$, as all gold evidence) and measure the resulting entropy distribution. Figure 7 reports this across four multi-hop QA benchmarks: HotPotQA, TriviaQA, MuSiQue, and 2WikiMultiHopQA, using model Qwen-2.5 Instruct at the 3B and 7B scales without any finetuning. Across all datasets and model combinations, H_k decreases monotonically as k increases. At $k=0$, the model exhibits high entropy, reflecting substantial uncertainty over the answer space; at $k=k_{\max}$, the distribution concentrates near $H_k < 1.0$. This separation is most pronounced on compositional multi-hop benchmarks, such as MuSiQue and 2WikiMultiHopQA, where the absence of bridge documents prevents shortcuts via parametric recall.

Observation 1 (Entropy-Sufficiency Correlation). H_k decreases monotonically in k ; in particular, $\mathbb{E}[H_k \mid \mathcal{P} \subseteq \mathcal{G}_k] < \mathbb{E}[H_k \mid \mathcal{P} \not\subseteq \mathcal{G}_k]$.

Implication for retrieval control. Observation 1 establishes that H_k is a reliable signal of evidence sufficiency. Moreover, H_k is a deterministic function of the same hidden state h_k used for retrieval, as in Eq. 1, so entropy-based stopping decisions can, in principle, be inferred from h_k alone, without requiring full text generation at each retrieval step. Building on this, Section 3 introduces a lightweight MLP-based Retrieval Control Head that operates directly on $h_{[\text{PRED}]}$ to predict whether further retrieval is needed.

D. Control-Head Supervision under Varying Annotation Regimes

The control-head label $y^{(r)}$ defined in Eq. 5 requires access to a set of gold supporting documents $\mathcal{P}$. Such annotations are directly available in standard multi-hop QA datasets (e.g., HotpotQA, 2WikiMultiHopQA, MuSiQue), and are commonly used by prior retrieval-augmented methods for supervision or reward design (Jin et al., 2025; Shi et al., 2025b; Trivedi et al., 2023). Our framework therefore operates under the same supervision assumptions as competitive baselines in the multi-hop setting.

Span-containment labeling for single-hop datasets. For single-hop datasets (e.g., NQ, TriviaQA), passage-level annotations are not directly aligned with the retrieval corpus. We adopt a standard Dense Passage Retrieval (DPR) (Karpukhin

et al., 2020) strategy to construct $\mathcal{P}$.

Chunking. We first segment the Wikipedia corpus $\mathcal{C}$ into fixed-length passages (100-word chunks), which serve as the retrieval units. This preprocessing step is deterministic and independent of model training.

Positive passage assignment. We then construct a single positive passage per query using dataset-specific heuristics:

(i) *Datasets with annotated contexts (e.g., NQ).* We align the original annotated gold passage to the closest matching chunk in $\mathcal{C}$ based on answer span overlap. The matched chunk is treated as the positive passage. Examples for which no alignment is found are discarded.

(ii) *Datasets without annotated contexts (e.g., TriviaQA).* We adopt distant supervision by retrieving passages using a sparse retriever (e.g., BM25, E5), and selecting the highest-ranked passage that contains the answer string as the positive passage. If no retrieved passage within the top- k results contains the answer, the example is removed.

This procedure yields weakly supervised "gold" passages that are not exhaustively annotated but have been shown to be effective in prior work. The resulting set $\mathcal{P}$ is used consistently for both retrieval training and control-head supervision, ensuring a unified notion of relevance without requiring additional manual annotation.

E. More Experimental Results

E.1. Retrieval Control Head Analysis

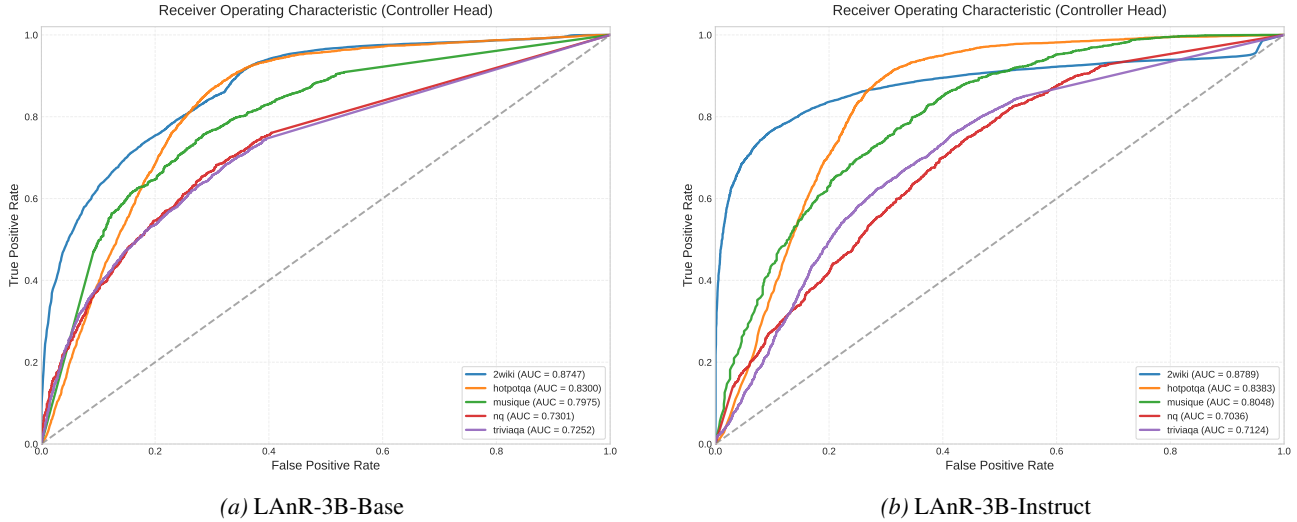


Figure 8. ROC curves for the retrieval control head across five benchmarks. Instruction tuning yields consistent gains, with the largest improvement on MuSiQue

Figure 8 shows ROC curves for the retrieval control head on both LAnR-3B-Base and LAnR-3B-Instruct across all five benchmarks. Both variants achieve strong AUC on multi-hop datasets, confirming that the $[\text{PRED}]$ hidden state carries a reliable evidence-sufficiency signal when supporting documents are compositionally distributed. Performance of both model variants is lower on single-hop datasets (Base: 0.730, 0.725; Instruct: 0.704, 0.712), where the distinction between “enough” and “not enough” evidence is less clear because a single retrieved document often suffices. The consistent improvement of the Instruct variant across all five datasets and the clear stratification by task complexity confirm that the control head learns a genuine sufficiency signal rather than a dataset-specific heuristic.

E.2. Effect of Model Size

Table 7 compares Search-R1, AutoRefine, and LAnR across Qwen-2.5-Base 3B and 7B backbones using both EM and F1 metrics. Scaling from 3B to 7B consistently improves performance for all methods, with the largest gains observed on multi-hop benchmarks that require compositional reasoning. At both model scales, LAnR achieves the strongest overall multi-hop performance. With the 3B backbone, LAnR-3B-Base attains the highest EM on HotpotQA (0.417), 2Wiki (0.402),

Table 7. EM and F1 across methods and model sizes (Qwen-2.5-Base, 3B and 7B) on five QA benchmarks.

Model	Metric	Single-Hop		Multi-Hop			Avg.
		NQ	TriviaQA	HotpotQA	2Wiki	MuSiQue	
<i>Qwen-2.5-3B-Base</i>							
SearchR1-3B-Base	EM	0.421	0.583	0.297	0.274	0.066	0.328
	F1	0.476	0.650	0.380	0.322	0.123	0.390
AutoRefine-3B-Base	EM	0.467	0.620	0.405	0.393	0.157	0.408
	F1	0.534	0.689	0.503	0.453	0.233	0.482
LAnR-3B-Base (ours)	EM	0.455	0.610	0.417	0.402	0.183	0.410
	F1	0.535	0.654	0.589	0.603	0.256	0.527
<i>Qwen-2.5-7B-Base</i>							
SearchR1-7B-Base	EM	0.469	0.627	0.410	0.272	0.173	0.390
	F1	0.552	0.700	0.517	0.327	0.236	0.466
AutoRefine-7B-Base	EM	0.484	0.659	0.451	0.405	0.187	0.437
	F1	0.574	0.729	0.573	0.607	0.283	0.553
LAnR-7B-Base (ours)	EM	0.482	0.631	0.459	0.415	0.205	0.438
	F1	0.570	0.724	0.590	0.612	0.295	0.558

 Table 8. Statistical analysis against search-during-think baselines. The p -value column represents the T-test result of LAnR-Instruct v.s. AutoRefine-Instruct and Search-R1-Instruct.

Model	NQ	TriviaQA	HotpotQA	2wiki	Musique	p -value
LAnR-Instruct	0.460 ± 0.008	0.613 ± 0.005	0.419 ± 0.006	0.408 ± 0.006	0.193 ± 0.005	—
AutoRefine-Instruct	0.461 ± 0.010	0.620 ± 0.007	0.405 ± 0.012	0.404 ± 0.010	0.145 ± 0.011	7.49×10^{-3}
Search-R1-Instruct	0.410 ± 0.009	0.597 ± 0.019	0.315 ± 0.016	0.254 ± 0.023	0.124 ± 0.005	1.31×10^{-39}

and MuSiQue (0.183), while also substantially outperforming prior methods in F1, particularly on HotpotQA (0.589 vs. 0.503 for AutoRefine) and 2Wiki (0.603 vs. 0.453). Similar trends hold at 7B scale, where LAnR-7B-Base achieves the best EM and F1 across all three multi-hop datasets, reaching 0.459/0.590 on HotpotQA, 0.415/0.612 on 2Wiki, and 0.205/0.295 on MuSiQue. Although AutoRefine remains competitive on single-hop datasets such as NQ and TriviaQA, the results show that LAnR benefits more consistently from increased model capacity, suggesting that latent retrieval scales effectively with stronger backbone representations and is particularly advantageous for complex multi-step reasoning tasks.

E.3. Variance Analysis

Table 8 reports the mean and standard deviation across three runs with different random seeds to evaluate the robustness of search-during-think methods. LAnR-Instruct consistently achieves lower variance than competing baselines across most datasets, indicating more stable retrieval and reasoning behavior. To assess statistical significance, we perform paired T-tests between LAnR-Instruct and AutoRefine-Instruct. The resulting low p -values ($p \ll 0.01$) confirm that the performance differences are statistically significant, demonstrating that the gains of latent retrieval are reliable rather than arising from random variation.

E.4. Ablation on Numbers of [PRED] Tokens

We further analyze the effect of training with multiple [PRED] tokens, detailed in Figure 9. Increasing the number of [PRED] tokens encourages the model to construct more abstract latent queries. On single-hop benchmarks such as TriviaQA and NQ, LAnR consistently benefits from additional [PRED] tokens, with performance improvements emerging early in training. In contrast, on more challenging multi-hop benchmarks, models with multiple [PRED] tokens require substantially longer training before surpassing the performance of the single-[PRED] setting. These results suggest that while additional [PRED] tokens can improve abstraction capability, they also increase optimization difficulty for complex reasoning tasks. Therefore, we adopt $N = 1$ as the default configuration throughout our experiments.

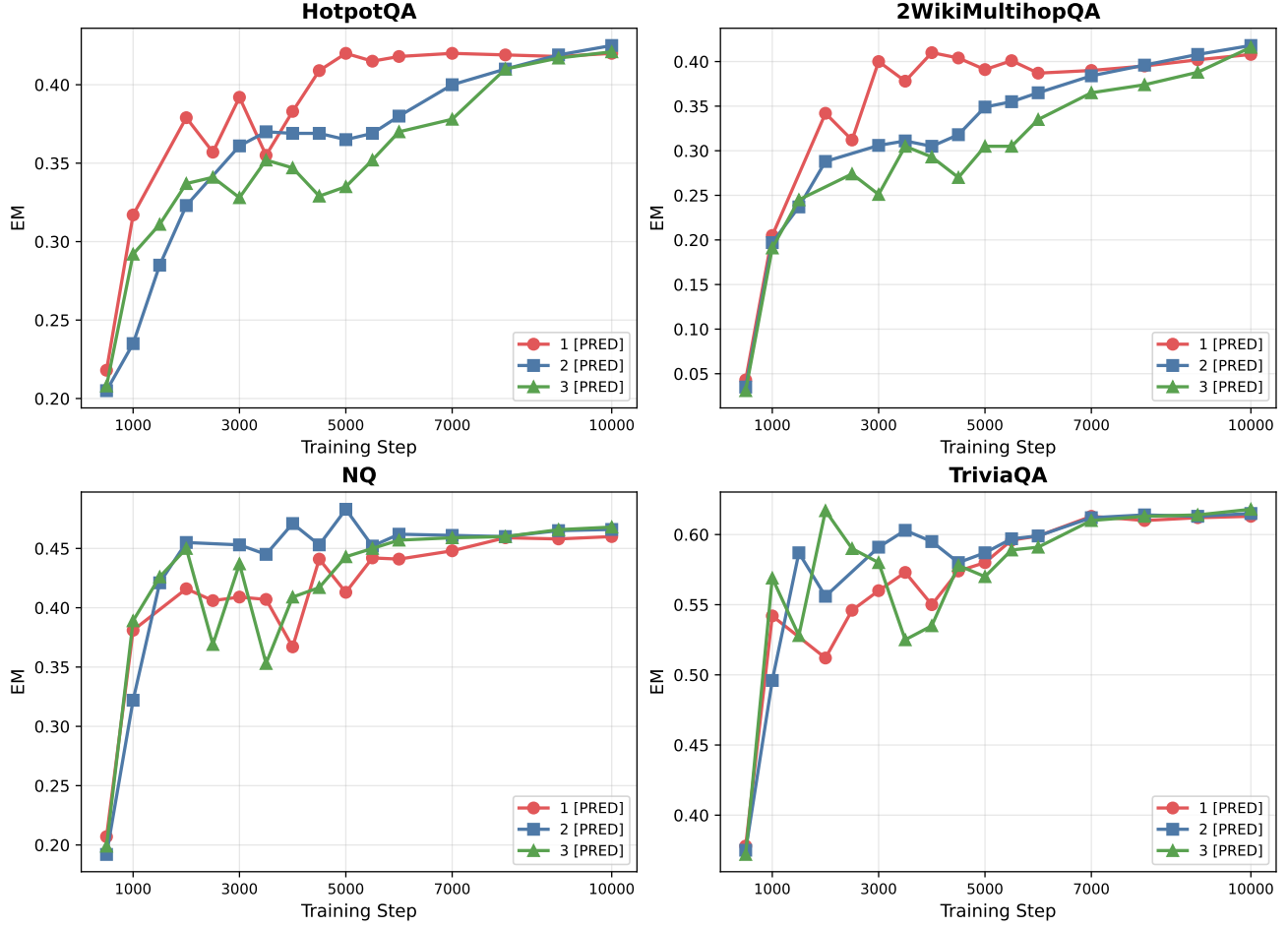


Figure 9. EM accuracy by training step for models trained with 1, 2, and 3 [PRED] tokens across four benchmarks. $N=1$ converges fastest and performs best on multi-hop datasets; additional tokens offer marginal gains only on single-hop TriviaQA.

F. Case Studies

The following examples trace LAnR’s retrieval process hop by hop. At each search, the model runs one forward pass over a growing context ending with [PRED]. The hidden state at that position encodes all evidence seen so far, drives the next dense retrieval query, and is fed to the MLP control head to decide whether to continue. The context after h searches is:

$$[Q] \text{ question } \underbrace{[d_1^{(1)}] \cdots [d_K^{(1)}]}_{\text{search 1}} \cdots \underbrace{[d_1^{(r)}] \cdots [d_K^{(r)}]}_{\text{search } r} [\text{PRED}]$$

Gold supporting documents are underlined with *.

Example 1 - MuSiQue (3-hop)

2WikiMultihopQA - 2-hop compositional

Question: "Where do Greyhound buses leave from in the city where the performer of Darlings formed?"

Reasoning chain: Darlings → performer: Kevin Drew → formation city: Toronto → bus terminal

Turn 1
 $p_{\text{ctrl}} = 0.9977 \rightarrow \text{CONTINUE}$
Context: [Q] Where do Greyhound buses leave from ...formed? [PRED]

Retrieved: Kevin Drew*, Spirit If ..., Detropia

[Kevin Drew] Kevin Drew (born September 9, 1976) is a Canadian musician and songwriter who, together with Brendan Canning, founded the expansive Toronto baroque-pop collective Broken Social Scene.

Turn 2
 $p_{\text{ctrl}} = 0.7922 \rightarrow \text{CONTINUE}$
Context: [Q] ... [Kevin Drew: ...Toronto baroque-pop collective...] [Spirit

If:...] [Detropia:...] [PRED]

Retrieved: Toronto Coach Terminal*, Alvarado Transportation Center, Dufferin St. Bridges

[Toronto Coach Terminal] The Toronto Coach Terminal is the central bus station for inter-city services in Toronto, Ontario, Canada. It is located at 610 Bay Street, in the city's Downtown.

Turn 3
 $p_{\text{ctrl}} \approx 0 \rightarrow \text{STOP}$
Context: [Q] ... [Kevin Drew:...] [Toronto Coach Terminal: ...Toronto...Bay St...] ... [PRED]

Retrieved: Darlings (Kevin Drew album)*, Peace Center, Darlington County

[Darlings (Kevin Drew album)] Darlings is the second solo album by Broken Social Scene co-founder Kevin Drew. It was released on March 18, 2014.

Answer: The Toronto Coach Terminal ✓ *Correct*

* gold document

The [PRED] state at Search 1 already encodes *Toronto* implicitly through the Kevin Drew article, letting the model jump directly to the bus-terminal query at Search 2 without materialising "Toronto" as an explicit intermediate result. All three gold documents are retrieved across three latent searches, and the control head stops as soon as the evidence chain is complete.

Example 2 - 2WikiMultihopQA (2-hop)

2WikiMultihopQA - 2-hop compositional

Question: "Where was the director of film *The Green Fog* born?"

Reasoning chain: The Green Fog → director: Guy Maddin → birthplace: Winnipeg, Manitoba

Turn 1

$p_{\text{ctrl}} = 0.9999 \rightarrow \text{CONTINUE}$

Context: [Q] Where was the director of film *The Green Fog* born? [PRED]

Retrieved: The Green Fog*, Walon Green, Adam Green (filmmaker)

[**The Green Fog**] *The Green Fog* is an experimental film directed by Guy Maddin, Evan Johnson, and Galen Johnson that loosely revisits the plot of Alfred Hitchcock's *Vertigo* through found footage.

Turn 2

$p_{\text{ctrl}} = 0.9992 \rightarrow \text{CONTINUE}$

Context: [Q] ... [*The Green Fog*: ...directed by Guy Maddin...] [Walon Green:...] [Adam Green:...] [PRED]

Retrieved: Guy Maddin*, *The Heart of the World*, *The Forbidden Room*

[**Guy Maddin**] Guy Maddin (born February 28, 1956) is a Canadian screenwriter, director, author, cinematographer, and film editor of both features and short films, from **Winnipeg, Manitoba**.

Turn 3

$p_{\text{ctrl}} = 0.0000 \rightarrow \text{STOP}$

Context: [Q] ... [*The Green Fog*:...] [Guy Maddin: ...*Winnipeg, Manitoba*...] ... [PRED]

(Control head halts; no new documents retrieved.)

Answer: Winnipeg, Manitoba ✓ *Correct*

* gold document

This example shows the sharpest control-head transition across all three datasets: p_{ctrl} holds at 0.9999 and 0.9992 while evidence accumulates, then drops to 0.0000 the moment *Guy Maddin* and his birthplace are both in context. No text query names "Guy Maddin" at any step; the latent query at Search 2 is shaped entirely by the [PRED] representation formed over the *The Green Fog* article.