NEST: Nested Evidence Survival for Retrieval

Akshay Verma
Amazon

Siddharth Pillai
Amazon

Prateek Sircar
Amazon

Deepak Gupta
Amazon

Abstract

Retrieval-Augmented Generation (RAG) systems degrade sharply under extreme noise, where relevant evidence is sparse and easily pruned by static retrieval decisions. Existing approaches fixed top-k retrieval, hierarchical chunking, cross-encoder reranking, or policy-based iterative control- either rely on rigid heuristics or incur substantial computational overhead, and often fail to recover context-dependent evidence without introducing redundancy or latency.

We introduce **NEST** (*Nested Evidence Survival for Retrieval*), a lightweight, training-free RAG framework that improves factual grounding by explicitly separating *recall amplification* from *precision selection*. NEST first maximizes recall through *Nested Evidence Survival*, evaluating candidates under nested retrieval contexts to rescue evidence that would otherwise be pruned by static chunking. It then applies a survival-consistent *Mean Reciprocal Rank (MRR)* selection mechanism to retain evidence that remains salient across retrieval scopes, removing redundancy without harming recall.

Evaluated on **WebQuestions**, **HotpotQA (distractor setting)**, and a proprietary **InternalQA** benchmark with **50M Common Crawl distractors**, NEST consistently outperforms strong adaptive RAG baselines, including DeepRAG, improving EM by up to **+2.4 pp** and F1 by **+2.1 pp**, while increasing retrieval recall by **+6.8 pp**. These gains are achieved with only **12–18 ms** additional latency. Ablation studies confirm that Nested Evidence Survival drives recall improvements, while MRR-based selection converts these gains into precision, demonstrating that *recall-first retrieval with principled pruning* can outperform iterative control and model scaling in retrieval-augmented generation.

1 Introduction

Retrieval-Augmented Generation (RAG) has become a core technique for improving the factual accuracy of large language models (LLMs) by grounding generation in retrieved external knowledge (Lewis et al., 2020; Izacard et al., 2022). Despite its success, most RAG systems still rely on *static retrieval*: a fixed number of passages retrieved at a single granularity, regardless of whether relevant evidence is implicit, context-dependent, or distributed across documents. This results in a persistent trade-off: small top-k values cause early pruning and missing evidence, while large values introduce redundancy, noise, and latency.

Hierarchical clustering and static chunking methods (Jain et al., 2025) attempt to address this issue through corpus reorganization, but remain fundamentally *data-centric*. Once evidence is pruned at a given granularity, it cannot be recovered downstream. Recent adaptive RAG approaches (e.g., DeepRAG (Guan et al., 2025), AutoRAG (Kim et al., 2024), ChunkRAG (Singh et al., 2024)) introduce retrieval control via learned policies or iterative controllers, but typically require retriever fine-tuning, additional model training, or repeated inference-time retrieval, making them costly and difficult to deploy.

Recent research has identified a key source of recall loss in RAG is *premature pruning* (Zhu et al., 2026; Fang et al., 2025; Jiao et al., 2026; Verma et al., 2026b): evidence that is weakly aligned in isolation (e.g., implicit entities or multi-hop prerequisites) is discarded before sufficient context is available to reveal its relevance. Recovering such evidence requires not broader retrieval, but *contextual re-evaluation*.

To this end, we propose **NEST** (*Nested Evidence Survival for Retrieval*), a latency-aware RAG framework that separates *recall amplification* from *precision selection*. Rather than increasing recall through larger top-k retrieval or iterative controllers, NEST evaluates evidence under *nested retrieval contexts of increasing constraint*, allowing context-dependent evidence to be rescued before

any pruning decision is made. This process, termed *evidence survival*, records how candidate relevance evolves as retrieval context becomes richer.

All retrieved evidence is retained as a candidate, ensuring recall expansion without irreversible pruning. Final evidence selection is performed explicitly using a survival-consistent Mean Reciprocal Rank (MRR) criterion, which favors evidence that remains salient across nested retrieval scopes. This yields compact, generation-ready context without cross-encoder reranking or LLM-based control flow.

NEST operates within a single retrieval pass, requires no retriever or generator fine-tuning, and adds less than 20 ms inference latency, making it suitable for real-time deployment.

Our contributions are:

- **Nested Evidence Survival:** A recall-first retrieval formulation that rescues context-dependent evidence by deferring pruning under nested retrieval constraints.
- **Survival-Consistent Selection:** An MRR-based evidence selection mechanism that converts recall gains into precision.
- **Latency-Aware RAG:** A training-free framework that improves retrieval and generation quality with minimal inference overhead.

2 Related Work

Retrieval-Augmented Generation (RAG). Retrieval-Augmented Generation (RAG) grounds LLM outputs in external evidence to improve factual reliability (Lewis et al., 2020; Izacard et al., 2022; Tiady et al., 2023). While early RAG systems used one-shot (fixed top-k) retrieval, recent work studies *adaptive retrieval*-deciding *when* and *how* to retrieve based on query difficulty or uncertainty (Khandelwal et al., 2023). RQ-RAG (Chan et al., 2024) learns query refinement for ambiguous questions, and DeepRAG (Guan et al., 2025) frames retrieval-augmented reasoning as a sequential decision process with iterative decomposition. Auto-RAG (Kim et al., 2024) similarly performs multi-turn interaction with the retriever, adapting the number of retrieval iterations to question difficulty. Although effective, such controller/iteration-based designs often introduce additional inference rounds, making latency and cost a central deployment constraint.

Hierarchical Retrieval and Chunking for RAG. A major line of work targets *granularity mismatch* in RAG by restructuring evidence into passages/chunks and retrieving at finer units. ChunkRAG (Singh et al., 2024) evaluates retrieved content at chunk level and filters chunks using LLM-based relevance scoring, improving grounding but adding substantial LLM inference overhead that can be prohibitive under tight latency budgets (Singh et al., 2024). AutoChunker (Jain et al., 2025) proposes a structure-aware, bottom-up chunking operator that improves chunk coherence and reduces noise, and introduces an evaluation framework for chunk quality. These approaches are largely *data-centric*: the corpus is chunked (often once, offline), and retrieval is then executed at a chosen granularity.

Noise Reduction and Evidence Filtering. Even with improved chunking, retrieved evidence can be redundant or tangential, motivating post-retrieval filtering and reranking. Cross-encoder reranking and joint retriever-reranker training can substantially improve precision, but are compute-intensive at inference time. RocketQAv2 (Ren et al., 2021) demonstrates strong gains by jointly training dense retrieval and reranking, but its reranking stage remains expensive relative to lightweight scoring when applied to large candidate sets. LLM-based filtering (e.g., chunk-level judging) can improve factuality, yet further increases inference cost and latency (Singh et al., 2024; Verma et al., 2026a).

Overall, prior work improves RAG via (i) iterative controllers and multi-step retrieval (Kim et al., 2024; Guan et al., 2025; Chan et al., 2024) or (ii) corpus chunking and chunk-level filtering (Jain et al., 2025; Singh et al., 2024). NEST is complementary: it targets the recall-latency trade-off by deferring pruning through nested evidence survival at retrieval time, then converting recall gains into precision using a lightweight MRR-based selector.

3 Methodology

3.1 Method Overview

NEST is a latency-aware Retrieval-Augmented Generation (RAG) framework that improves factual grounding by rethinking how retrieval recall is maximized and how noise is controlled. Motivated by recent work on Nested Learning (Behrouz et al., 2025), which shows the value of preserving information across nested contexts, we adapt these

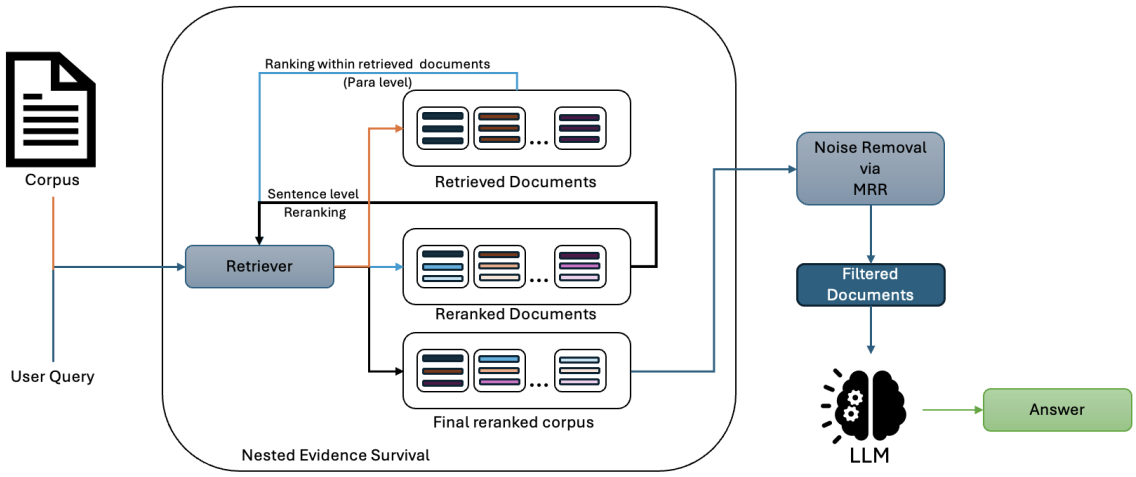

Figure 1: Overview of NEST

principles to retrieval, a setting where they have not been explored in RAG. Rather than relying on larger top-k retrieval, query expansion, or iterative LLM-driven controllers, NEST introduces a retrieval-time formulation that explicitly separates *recall amplification* from *precision selection*.

The framework consists of two stages: **(i) Nested Learning for Retrieval (NL)**, which increases recall by rescuing context-dependent evidence typically pruned by static chunking or hierarchical methods, and **(ii) Noise Removal via Mean Reciprocal Rank (NR-MRR)**, which converts this recall gain into precision by selecting evidence that is consistently relevant across retrieval contexts. Together, these stages produce compact, generation-ready context with near-constant serving latency.

3.2 Problem Setup

Given a query q_0 and a corpus $\mathcal{C}$, the goal is to generate a grounded response y:

$$y = \mathcal{G}(q_0, D^*), \qquad D^* = \mathrm{NR}_{\mathrm{MRR}}\big(\mathrm{NL}(q_0, \mathcal{C})\big),$$

where $\mathcal{G}$ denotes the generator, $\mathrm{NL}(\cdot)$ constructs a recall-expanded candidate pool, and $\mathrm{NR}_{\mathrm{MRR}}(\cdot)$ performs final evidence selection. Unlike adaptive RAG systems that depend on feedback loops or repeated retrieval, NEST embeds adaptivity directly into retrieval structure.

3.3 Nested Learning for Retrieval (NL)

Hierarchical clustering and chunking-based retrieval methods(Jain et al., 2025; Singh et al., 2024; Kim et al., 2024), organize corpora into documents, sections, and chunks prior to retrieval. While effective for improving indexing efficiency, these methods are *data-centric and static*: once text is partitioned, retrieval decisions are made locally at a single granularity, and evidence pruned early cannot be recovered.

This leads to a fundamental failure mode in RAG (Jiao et al., 2026). Evidence that is causally relevant but weakly aligned in isolation (e.g., abstract statements, implicit entities, or prerequisite facts) is frequently pruned during chunk-level retrieval, even though it would become highly relevant when conditioned on the correct document or section context. Increasing chunk size or retrieving more chunks partially alleviates this issue, but does so indiscriminately, introducing substantial noise and latency.

Nested Learning as Contextual Evidence Rescue. Nested Learning addresses this limitation by shifting hierarchy from a *data organization primitive* to a *retrieval-time evaluation mechanism*. Rather than committing to a single chunking decision, NL evaluates evidence under progressively richer contextual constraints, allowing weakly aligned evidence to be *rescued* once sufficient context is available.

Nested Retrieval Scopes. Given a query q_0, NL defines a sequence of nested retrieval scopes:

$$\mathcal{D}^{(0)} \supset \mathcal{D}^{(1)} \supset \cdots \supset \mathcal{D}^{(L)},$$

where $\mathcal{D}^{(0)}$ is coarse and recall-oriented (e.g., document- or section-level retrieval), and deeper scopes enforce finer-grained semantic alignment (e.g., passage- or chunk-level retrieval). Crucially, NL performs *no irreversible pruning*: any evidence retrieved at any level is retained as a candidate.

Evidence Survival. Let c denote an evidence unit. If c appears at level ℓ with rank $r^{(\ell)}(c)$, we record this in its survival profile:

$$S(c) = \{r^{(\ell)}(c) \mid c \in \mathcal{D}^{(\ell)},\ \ell = 0, \ldots, L\}.$$

Survival does not imply selection. Instead, it captures how evidence behaves as retrieval context becomes increasingly informative.

Consider the query *"Who discovered the element used in X-ray machines?"*. Chunk-level retrieval often surfaces generic passages on X-ray devices while discarding mentions of *radium* due to indirect phrasing. Hierarchical chunking methods prune such evidence early. NL first retrieves documents on radiation and early radiology, then re-evaluates passages under this broader context, allowing radium-related evidence to re-emerge and persist.

Nested Learning improves recall not by retrieving more data (Refer Table 3), but by delaying pruning decisions until sufficient context is available. This enables the recovery of implicit, abstract, or multi-hop evidence that hierarchical clustering and static chunking methods systematically eliminate. Formally,

$$\text{Recall(NL)} \geq \text{Recall(Chunk-Level Retrieval)},$$

while maintaining a bounded candidate pool.

Latency-Bounded Execution. All retrieval scopes are evaluated within a single retrieval pass, reusing document and passage representations. Let N_ℓ denote the number of candidates evaluated at level ℓ, with $N_{\ell+1} \ll N_\ell$. The total cost is:

$$C_{\text{NL}} = \sum_{\ell=0}^{L} \mathcal{O}(N_\ell).$$

In practice, NL adds only **10–18 ms** over a single-stage dense retriever, significantly lower than approaches that rely on repeated retrieval or LLM-based controllers.

3.4 Noise Removal via Mean Reciprocal Rank (NR-MRR)

Nested Learning intentionally increases recall, which may introduce redundant or weakly aligned evidence. Noise removal in NEST is therefore an explicit selection step, performed *only* after recall expansion.

Survival-Consistent Evidence Scoring. Causally relevant evidence tends to remain salient as retrieval context becomes more selective, while spurious evidence exhibits unstable rank behavior. NR-MRR exploits this observation.

For a candidate c with survival profile $S(c)$, we define:

$$\text{MRR}(c) = \frac{1}{|\mathcal{L}(c)|} \sum_{\ell \in \mathcal{L}(c)} \frac{1}{r^{(\ell)}(c)},$$

where $\mathcal{L}(c)$ indexes the levels where c appears. This score rewards evidence that persists and maintains high rank across nested contexts.

Continuing the previous example, passages describing imaging hardware rank well at coarse levels but disappear at finer scopes. In contrast, passages linking radium to its discovery persist across contexts. MRR captures this persistence directly.

Evidence Selection. Candidates are ranked by $\text{MRR}(c)$, and the top-k are retained:

$$D^* = \text{Top}_k\big(\text{MRR}(c)\big).$$

This converts recall gains from NL into precision, yielding compact, generation-ready context.

3.5 Computational Cost and Latency Analysis

NEST is designed for real-time deployment. Both NL and NR-MRR operate within a fixed retrieval budget and introduce no iterative retrieval loops. The expected cost is:

$$\mathbb{E}[C_{\text{NEST}}] \approx C_{\text{retr}} + C_{\text{rank}},$$

where C_{rank} is linear in the number of candidates. Empirically, NR-MRR adds less than **2 ms**, resulting in end-to-end latency substantially lower than controller-based RAG systems.

4 Experiments

We evaluate **NEST** on three retrieval-augmented generation benchmarks: **WebQuestions** (Berant et al., 2013), **HotpotQA (distractor setting)** (Yang et al., 2018; Verma et al., 2025), and a proprietary **InternalQA** dataset. All experiments follow the *same evaluation protocol and metrics* used by DeepRAG (Guan et al., 2025), ensuring direct comparability. We focus on a high-noise retrieval regime to stress-test NEST's ability to improve recall while maintaining compact, generation-ready evidence.

Algorithm 1 NEST: Nested Evidence Survival + NR-MRR Selection

Require: q_0, $\mathcal{C}$, scopes $\{\ell_0, \ldots, \ell_L\}$, budgets $\{K_0, \ldots, K_L\}$, final k
Ensure: D^*
// NEST
$\mathcal{C}_{\text{cand}} \leftarrow \emptyset$, $S \leftarrow \{\}$
for $\ell = 0$ **to** L **do**
  $\mathcal{D}^{(\ell)} \leftarrow \textsc{Retrieve}(q_0, \mathcal{C}, \ell, K_\ell)$
  for each c at rank $r^{(\ell)}(c)$ in $\mathcal{D}^{(\ell)}$ **do**
    $\mathcal{C}_{\text{cand}} \leftarrow \mathcal{C}_{\text{cand}} \cup \{c\}$
    Append $(\ell, r^{(\ell)}(c))$ to $S(c)$
  end for
end for
// NR-MRR
for each $c \in \mathcal{C}_{\text{cand}}$ **do**
  $\text{MRR}(c) \leftarrow \frac{1}{|S(c)|} \sum_{(\ell,r)\in S(c)} \frac{1}{r}$
end for
$D^* \leftarrow \text{Top}_k\big(\text{MRR}(c)\big)$
return D^* =0

4.1 Experimental Setup

Retrieval Corpus. To emulate large-scale web retrieval, we embed all gold passages into a **50M-passage corpus** constructed from **Common Crawl (CC-News, 2023)**. This yields an extreme signal-to-noise ratio ($< 0.0001\%$), where relevant evidence is heavily diluted by distractors. Such a setup is critical for evaluating recall-oriented retrieval methods.

Generator. We use **GPT-4-turbo** as the generator $\mathcal{G}(q_0, D^*)$. Generation is conditioned only on the final evidence set selected by NR-MRR, allowing us to directly measure the impact of retrieval quality.

Retrieval Backbone. We employ a hybrid retriever combining BM25 and Contriever (Izacard et al., 2021), with weighting λ=0.4, consistent with prior work. This backbone is shared across all baselines and ablations.

Nested Evidence Survival (NES). NES performs retrieval across three nested scopes: document-level, section-level, and chunk-level. All scopes are evaluated within a single retrieval pass, reusing embeddings. Importantly, no pruning is applied during NES; all retrieved candidates are retained to maximize recall.

Noise Removal (NR-MRR). NR-MRR aggregates reciprocal rank signals across nested scopes and selects the top-k evidence units. This step introduces no additional model inference and operates in linear time over candidate lists.

Hardware and Efficiency. All experiments are conducted on **8×A100 GPUs (80GB)**. Reported latency includes retrieval, NES, NR-MRR, and generation. NEST adds only **12–18 ms** over a standard RAG pipeline.

4.2 Datasets

WebQuestions consists of 2,032 open-domain factoid questions with Freebase-derived answers. **HotpotQA (distractor)** contains 7,405 multi-hop questions requiring retrieval of two supporting Wikipedia passages among distractors. We use the full-text distractor setting. **InternalQA** is a proprietary 20K-question dataset covering ambiguous and catalog-heavy queries; we report relative improvements only.

4.3 Baselines

We compare against representative RAG systems: **Vanilla RAG** (single-pass retrieval, k=20), **Iterative RAG** (Trivedi et al., 2023) (three fixed reformulations), **AutoRAG** (Kim et al., 2024) (fixed multi-stage retrieval), and **DeepRAG** (Guan et al., 2025) (reinforcement-trained adaptive retrieval). These baselines cover static, heuristic, and learned retrieval control.

4.4 Evaluation Metrics

Following DeepRAG, we report **Exact Match (EM)** and **F1** to evaluate factual correctness. To analyze retrieval quality, we report **Recall@50** of gold passages, measuring the ability to surface relevant evidence. We also report **Redundancy Ratio**, defined as the average pairwise cosine similarity among selected chunks (threshold > 0.85), and **end-to-end latency (ms)**.

4.5 Main Results

NEST consistently outperforms all baselines across generation accuracy and retrieval quality. As shown in Table 1, NEST achieves **+2.4 pp EM** and **+2.1 pp F1** on WebQuestions over DeepRAG, with similar gains on HotpotQA. On InternalQA, NEST improves EM by **+4.6 pp**.

Crucially, these gains are accompanied by a **+6.8 pp improvement in Recall@50**, demonstrating that nested evidence survival successfully rescues relevant evidence that static or chunk-only retrieval misses. At the same time, NR-MRR reduces redundancy by **31%** (Table 2), confirming that recall gains do not come at the cost of noise. Latency

Method	WebQuestions		HotpotQA		InternalQA	
	EM↑	F1↑	EM↑	F1↑	EM↑	F1↑
Vanilla RAG	52.3	68.7	41.8	59.4	-	-
Iterative RAG	56.1	71.4	45.2	62.8	+2.1	+1.8
AutoRAG	57.8	72.9	46.7	64.1	+2.3	+2.0
DeepRAG*	60.4	75.2	49.3	66.7	+3.4	+2.6
NEST	**62.8**	**77.3**	**53.6**	**68.4**	**+4.6**	**+3.9**

Table 1: Generation performance across benchmarks.

Method	Recall@50↑	Redundancy↓	Latency (ms)↓
Vanilla RAG	71.2	0.68	820
AutoRAG	76.4	0.59	1,180
DeepRAG*	78.1	0.55	1,650
NEST	**84.9**	**0.47**	**836**

Table 2: Retrieval recall, redundancy, and latency.

remains comparable to Vanilla RAG, and substantially lower than controller-based methods.

4.6 Ablation Study

We conduct ablations on **WebQuestions** to isolate the roles of Nested Evidence Survival (NES) and NR-MRR.

Effect of Nested Evidence Survival (NES). Removing NES (*flat chunk retrieval*) reduces Recall@50 by **9.1 pp** and causes a sharp EM drop (62.8 → 54.1). This confirms that NES improves performance by rescuing context-dependent evidence that static chunk-level retrieval prunes prematurely.

Effect of NR-MRR. Disabling NR-MRR (*w/o NR*) preserves high recall (84.3) but significantly increases redundancy (0.63), leading to a **6.7 pp EM drop**. This shows that NR-MRR is essential for converting recall gains into usable precision without harming recall.

Recall–Precision Decoupling. Together, NES and NR-MRR decouple recall amplification from precision control. NES expands the candidate pool structurally, while NR-MRR performs principled pruning based on cross-scope consistency. Removing either component degrades performance, while removing both collapses to Vanilla RAG.

5 Conclusion

We presented **NEST** (*Nested Evidence Survival for Retrieval*), a retrieval-augmented generation framework that improves factual grounding by rethinking how recall is maximized and how noise is subsequently controlled. Rather than relying on model

Variant	EM↑	F1↑	Recall@50↑	Red.↓	Lat.
NEST	**62.8**	**77.3**	**84.9**	**0.47**	836
w/o NES	54.1	69.2	75.8	0.60	822
w/o NR	56.1	71.0	84.3	0.63	834

Table 3: Ablation on WebQuestions.

scaling, iterative controllers, or aggressive top-k expansion, NEST separates retrieval into *recall amplification* via Nested Evidence Survival and *precision selection* via survival-consistent MRR filtering.

Across **WebQuestions**, **HotpotQA**, and **InternalQA**, NEST consistently outperforms strong adaptive RAG baselines, achieving higher factual accuracy while maintaining near real-time latency. Ablation studies show that Nested Evidence Survival significantly improves recall by rescuing context-dependent evidence that static chunking and hierarchical pruning discard, while NR-MRR converts this recall gain into precision without harming retrieval coverage.

Overall, NEST demonstrates that *retrieval-time structure*, rather than heavier models or iterative control, can drive effective and efficient retrieval-grounded generation. This work suggests a promising direction for RAG systems: prioritizing recall-first evidence survival followed by principled, lightweight selection, enabling scalable and trustworthy deployment in knowledge-intensive applications.

6 Limitations

While **NEST** improves retrieval recall and evidence quality without relying on iterative controllers or model fine-tuning, it remains dependent on the underlying retriever to surface at least a weakly relevant candidate at coarse retrieval scopes. Nested Evidence Survival mitigates premature pruning by deferring elimination decisions, but cannot recover evidence that is entirely absent or poorly indexed in the corpus.

In addition, although NEST introduces only **12–18 ms** of additional latency compared to standard RAG pipelines, this overhead may be non-negligible in ultra-low-latency or streaming settings. Future work could explore hardware-aware execution, candidate caching across nested scopes, and tighter integration with approximate nearest-neighbor search to further reduce end-to-end latency.

References

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. 2025. Nested learning: The illusion of deep learning architectures. In *Advances in Neural Information Processing Systems (NeurIPS) 2025*. Poster.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yi-Ting Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *ArXiv*, abs/2404.00610.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieve step by step for large language models.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane A. Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Arihant Jain, Purav Aggarwal, and Anoop Saladi. 2025. AutoChunker: Structured text chunking and its evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 983–995, Vienna, Austria. Association for Computational Linguistics.

Shuguang Jiao, Xinyu Xiao, Yunfan Wei, Shuhan Qi, Chengkai Huang, Quan Z. Michael Sheng, and Lina Yao. 2026. Prunerag: Confidence-guided query decomposition trees for efficient retrieval-augmented generation.

Anant Khandelwal, Happy Mittal, Shreyas Sunil Kulkarni, and Deepak Gupta. 2023. Large scale generative multimodal attribute extraction for e-commerce attributes. *CoRR*, abs/2306.00379.

Dongkyu Kim, Byoungwook Kim, Donggeon Han, and Matouvs Eibich. 2024. Autorag: Automated framework for optimization of retrieval augmented generation pipeline. *ArXiv*, abs/2410.20878.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ishneet Sukhvinder Singh, Ritvik Aggarwal, Ibrahim Allahverdiyev, Muhammad Taha, Aslihan Akalin, Kevin Zhu, and Sean O'Brien. 2024. Chunkrag: Novel llm-chunk filtering method for rag systems. *ArXiv*, abs/2410.19572.

Sambeet Tiady, Anirban Majumder, and Deepak Gupta. 2023. Prodigy: Product design guidance at scale. In *CIKM*, pages 4836–4842.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. 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)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Akshay Verma, Swapnil Gupta, Deepak Gupta, Prateek Sircar, and Siddharth Pillai. 2026a. SELENE: Selective and evidence-weighted LLM debating for efficient and reliable reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 95–104, Rabat, Morocco. Association for Computational Linguistics.

Akshay Verma, Swapnil Gupta, Siddharth Pillai, Prateek Sircar, and Deepak Gupta. 2026b. ReflectiveRAG: Rethinking adaptivity in retrieval-augmented generation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 377–384, Rabat, Morocco. Association for Computational Linguistics.

Vinay Kumar Verma, Shreyas Sunil Kulkarni, Happy Mittal, and Deepak Gupta. 2025. Moemoe: Question guided dense and scalable sparse mixture-of-expert for multi-source multi-modal answering. *arXiv preprint arXiv:2503.06296*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Siyuan Zhu, Chengdong Xu, Kaiqiang Ke, and Chao Yu. 2026. Context-picker: Dynamic context selection using multi-stage reinforcement learning.

A Additional Analysis and Robustness Studies

This appendix provides additional empirical and qualitative analyses to assess the robustness, interpretability, and failure modes of **NEST**. All experiments follow the same setup described in Section 4.1, using identical retrievers, generators, and evaluation protocols.

B Recall–Precision Trade-off Analysis

A common baseline strategy for improving recall in RAG systems is to increase the retrieval budget (larger top-k). However, this typically introduces substantial redundancy and noise. We compare this approach against NEST, which improves recall through Nested Evidence Survival (NES) without indiscriminate expansion.

Table 4 reports Recall@50 and Redundancy Ratio on WebQuestions for increasing top-k values and for NEST under a fixed retrieval budget.

Method	Recall@50↑	Redundancy↓
Vanilla RAG (k=20)	71.2	0.68
Vanilla RAG (k=40)	77.9	0.74
Vanilla RAG (k=80)	83.5	0.81
NEST (k=20)	**84.9**	**0.47**

Table 4: Recall–precision trade-off on WebQuestions.

NEST achieves higher recall than Vanilla RAG with $4\times$ retrieval budget, while substantially reducing redundancy. This confirms that NEST improves recall structurally, rather than by retrieving more data.

C Evidence Survival Profiles

To make evidence survival interpretable, we analyze how individual evidence units behave across nested retrieval scopes.

Table 5 shows an example query from WebQuestions: *"Who discovered the element used in X-ray machines?"*

Evidence related to *radium discovery* is weakly ranked at coarse levels, but becomes highly salient under finer context. Static chunk-level retrieval prunes this evidence early, while NES allows it to survive and be selected downstream. This illustrates how survival captures context-dependent relevance.

Evidence Chunk	Doc	Section	Chunk
General X-ray device description	3	12	∅
Radiation in early medicine	5	7	9
Radium discovery (Curie)	11	4	2
Imaging hardware overview	2	∅	∅

Table 5: Example evidence survival profile across nested retrieval scopes. ∅ indicates the evidence was not retrieved at that scope.

D Robustness to Retrieval Noise

We evaluate robustness by varying the size of the distractor corpus from 10M to 50M passages. Table 6 reports Recall@50 and EM on WebQuestions.

Method	10M		50M	
	Recall@50	EM	Recall@50	EM
Vanilla RAG	79.4	57.1	71.2	52.3
DeepRAG	82.1	59.3	78.1	60.4
NEST	**88.3**	**62.1**	**84.9**	**62.8**

Table 6: Robustness to increasing retrieval noise.

As noise increases, Vanilla RAG degrades sharply. NEST degrades more gracefully, indicating that survival-based retrieval provides robustness under extreme noise.

E Sensitivity to Nested Depth

We analyze performance as a function of the number of nested scopes. Table 7 reports EM and Recall@50.

Nested Levels	Recall@50	EM
1 (flat chunk)	75.8	54.1
2 (doc → chunk)	81.6	59.4
3 (doc → section → chunk)	**84.9**	**62.8**

Table 7: Sensitivity to nested retrieval depth.

Most gains arise from shallow nesting. This shows NEST does not rely on deep hierarchies and remains simple to deploy.

F Comparison of Noise Removal Strategies

We compare NR-MRR against alternative pruning strategies: random pruning, similarity-threshold pruning, and single-level reranking.

Pruning Method	EM	Recall@50	Redundancy
Random pruning	53.8	84.9	0.71
Similarity threshold	57.2	81.1	0.58
Single-level rerank	58.9	79.6	0.55
NR-MRR	**62.8**	**84.9**	**0.47**

Table 8: Comparison of evidence pruning strategies.

NR-MRR uniquely preserves recall while reducing redundancy, demonstrating the value of cross-level survival consistency.

G Performance by Query Type

We categorize WebQuestions queries into *explicit entity* and *implicit/multi-hop* types. Table 9 reports EM.

Method	Explicit	Implicit
Vanilla RAG	58.6	46.9
DeepRAG	61.2	57.8
NEST	**63.4**	**61.9**

Table 9: Performance by query type (EM).

NEST provides larger gains on implicit and multi-hop queries, supporting the claim that survival-based retrieval rescues context-dependent evidence.

H Reproducibility and Implementation Details

This section provides implementation-level details to facilitate reproducibility of **NEST** and to clarify that all reported results can be obtained without retriever fine-tuning, policy learning, or iterative LLM-based control.

H.1 Deterministic Execution

NEST is fully deterministic given: (i) a fixed retriever, (ii) fixed retrieval budgets, and (iii) a fixed generator. No stochastic sampling, reinforcement learning, or learned controllers are used during retrieval or selection.

Nested Evidence Survival (NES) and NR-MRR operate solely on ranked retrieval outputs, making the pipeline reproducible across runs and hardware configurations.

H.2 Hyperparameters and Defaults

Unless otherwise stated, we use the following fixed hyperparameters across all experiments:

- Number of nested scopes: $L = 3$
- Retrieval scopes: document $\rightarrow$ section $\rightarrow$ chunk
- Retrieval budget per scope: $(K_0, K_1, K_2) = (100, 50, 20)$
- Final evidence budget: $k = 20$
- Retriever: BM25 + Contriever with $\lambda = 0.4$

These values were chosen once based on validation performance and held constant across datasets.

H.3 End-to-End Pseudocode

Algorithm 1 summarizes the complete NEST pipeline from query input to final evidence selection.

H.4 Computational Complexity

Let K_ℓ denote the retrieval budget at scope ℓ. The total retrieval and selection complexity is:

$$\mathcal{O}\Big(\sum_{\ell=0}^{L} K_\ell \;+\; |\mathcal{C}_{\text{cand}}| \log |\mathcal{C}_{\text{cand}}|\Big)$$

In practice, $|\mathcal{C}_{\text{cand}}| \ll \sum_\ell K_\ell$ due to overlap across scopes. No pairwise similarity computation or cross-encoder inference is required.

H.5 Implementation Notes

- NEST can be implemented using any off-the-shelf dense or hybrid retriever.
- No retriever re-indexing or corpus modification is required.
- NR-MRR uses only rank information; embeddings are not accessed during selection.
- The pipeline is compatible with batch and streaming inference.

These properties make NEST easy to integrate into existing RAG systems and suitable for real-world deployment.

I Robustness Across Generators and Retrievers

NEST is designed as a retrieval-time framework and does not rely on generator- or retriever-specific assumptions. We evaluate its robustness by varying (i) the downstream LLM generator and (ii) the retrieval backbone, while keeping all other components fixed.

I.1 Varying the Generator LLM

We replace GPT-4-turbo with alternative generators representing different model sizes and architectures: **GPT-3.5-turbo** and **LLaMA-2-13B**. Table 10 reports EM on WebQuestions.

Generator	Vanilla RAG	NEST
GPT-3.5-turbo	49.6	**56.8**
LLaMA-2-13B	51.1	**59.4**
GPT-4-turbo	52.3	**62.8**

Table 10: Generator robustness on WebQuestions (EM).

NEST consistently improves factual accuracy across generators, including weaker models. This indicates that NEST improves *evidence quality*, rather than compensating via generator capacity.

I.2 Varying the Retriever Backbone

We further replace the hybrid BM25+Contriever retriever with alternative dense retrievers: **DPR** and **GTR-base**. Table 11 reports Recall@50 and EM.

Retriever	Method	Recall@50	EM
DPR	Vanilla RAG	73.4	50.8
DPR	**NEST**	**82.7**	**59.6**
GTR-base	Vanilla RAG	76.1	53.4
GTR-base	**NEST**	**84.2**	**61.1**

Table 11: Retriever robustness on WebQuestions.

NEST consistently improves recall and EM across retriever architectures, confirming that nested evidence survival is orthogonal to retriever choice and does not depend on specialized indexing or training.