QUID: Queries Unmasked by Iterative Diffusion

Noise Is All You Need: Diffusion-Based Query Expansion for Dense Retrieval

Yuvraj Jha[1], Lavanya Gupta[1], Amardiya S. Mujeeb[1], and Shreyas S[1]

[1]School of Computer Science and Engineering (SCOPE), VIT-AP University

Working draft · Month 4 in progress · Preprint coming soon

Guidance: Dr. G. Muneeswari, Professor (Grade 2), Head of the Department of Data Science and Engineering (DSE), School of Computer Science and Engineering (SCOPE), VIT-AP University.

Abstract

Dense retrieval suffers from query-document asymmetry, particularly in specialized domains like medicine and finance where autoregressive LLMs often hallucinate or fail to capture niche terminology. We propose QUID (Queries Unmasked by Iterative Diffusion), which frames query expansion as a denoising problem: treating user queries as "noisy" compressed representations of ideal documents, then using masked text diffusion to expand them. Unlike HyDE's "creative writer" approach that generates hypothetical facts, QUID acts as an "editor" that refines queries to their semantic essence—crucial for domains where precision trumps creativity.

Through rigorous evaluation on four BEIR benchmark datasets (1,321 queries), we establish domain-dependent effectiveness: QUID achieves +7.5% nDCG@10 on medical (NFCorpus, $p < 0.001$) and +6.5% on financial (FiQA, $p < 0.001$) domains, significantly outperforming HyDE. Conversely, HyDE wins on scientific claims (SciFact, +7.0%, $p = 0.005$) where factual knowledge matters more than precision. Our qualitative analysis reveals a surprising finding: QUID generates less vocabulary than HyDE yet achieves better results on specialized domains, suggesting the mechanism is embedding space regularization rather than vocabulary expansion.

Because always-expand is suboptimal when effectiveness is query-dependent, we further introduce an agentic query router that treats expansion methods as tools (vanilla / QUID / HyDE), critiques unsupervised retrieval confidence, and optionally retries. Early pilots on SciFact (n=50) show multi-step tool use reaching 0.6625 nDCG@10 vs. 0.6299 vanilla and 0.5999 always-QUID; larger confirmation runs and a learned router remain in progress. We release our implementation at https://github.com/Zhreyu/quid.

Keywords: Dense Retrieval, Query Expansion, Diffusion Language Models, Agentic Retrieval, BEIR Benchmark

Status: Capstone working draft (Months 1–3 complete; Month 4 agentic extension in progress). Not a published preprint.

1 Introduction

Dense retrieval systems embed queries and documents into a shared vector space, retrieving documents closest to the query embedding. A fundamental challenge is the query-document asymmetry: user queries are typically short (2–5 words) and underspecified, while relevant documents are longer and richer in detail. This creates an embedding space mismatch that hurts retrieval accuracy.

Query: “heart problems in elderly patients”
Step 0: heart problems in elderly patients [M][M][M]...
Step 8: heart problems in elderly patients cardiovascular [M][M]...
Step 16: heart problems elderly patients cardiovascular conditions cardiac [M]...
Step 32: heart problems elderly patients cardiovascular conditions cardiac disease myocardial risk factors treatment

Figure 1: Iterative unmasking in QUID. The query is progressively expanded through confidence-based token selection over 32 diffusion steps.

Prior work addresses this through query expansion using pseudo-relevance feedback [8] or hypothetical document generation [3]. HyDE (Hypothetical Document Embeddings) uses autoregressive LLMs like GPT-3.5 to generate hypothetical documents, then embeds these instead of the raw query. While effective, this approach has a critical limitation: autoregressive LLMs may hallucinate facts that don't exist in the target corpus, particularly in specialized domains where the LLM lacks expertise.

We propose QUID (Queries Unmasked by Iterative Diffusion), which frames query expansion as a denoising problem. We treat the user query as a “noisy” compressed representation of an ideal document and use masked text diffusion to “denoise” it into a fuller representation. This is inspired by recent advances in discrete diffusion for text generation [1, 6, 7].

Our key insight is the distinction between two expansion strategies:

- HyDE (“Creative Writer”): Generates hypothetical facts. Many words, some wrong. Wins when general knowledge helps.
- QUID (“Editor”): Refines semantic representation. Few words, high precision. Wins when precision trumps creativity.

Through evaluation on four BEIR datasets, we establish:

1. Domain-dependent effectiveness: QUID excels on specialized corpora (medical +7.5%, finance +6.5%); HyDE wins on general scientific text (+7.0%)
2. The “Editor” mechanism: QUID generates less vocabulary than HyDE yet wins—suggesting embedding space regularization rather than vocabulary expansion
3. Negative results that matter: Confidence gating and hybrid BM25+dense approaches fail, providing evidence for when diffusion helps vs. hurts
4. Agentic routing (in progress): When always-expand hurts, a multi-step observe→decide→act controller that calls expansion tools can recover accuracy (early SciFact / NFCorpus pilots)

2 Related Work

Dense Retrieval. Dense retrieval uses neural encoders to embed queries and documents [4, 5]. Modern models like BGE-M3 [2] achieve strong performance but still suffer from query-document asymmetry.

Query Expansion. Classical query expansion uses pseudo-relevance feedback (RM3) [9] or corpus statistics. HyDE [3] uses LLMs to generate hypothetical documents, achieving significant gains on multiple benchmarks. Query2Doc [12] takes a similar approach. Our work differs by using diffusion models rather than autoregressive LLMs.

Algorithm 1: QUID Query Expansion

Input: Query q, model M, steps S, gen_length L

$x \leftarrow \text{tokenize}(q) \oplus [\text{MASK}]^L$
blocks $\leftarrow L/\text{block\_length}$
for each block b do
  for step $s = 1$ to S/blocks do
    logits $\leftarrow M(x)$
    $\hat{x} \leftarrow \arg\max(\text{logits} + \text{GumbelNoise})$
    conf $\leftarrow \text{softmax}(\text{logits})[\hat{x}]$
    $k \leftarrow \text{tokens\_to\_unmask}(s)$
    idx $\leftarrow \text{top}_k(\text{conf})$
    $x[\text{idx}] \leftarrow \hat{x}[\text{idx}]$ // Unmask top-k
return decode(x)

Figure 2: QUID Query Expansion Algorithm

Text Diffusion Models. Discrete diffusion for text has seen rapid advances: D3PM [1], MDLM [10], SEDD [6], and LLaDA [7]. LLaDA particularly rivals LLaMA-3 8B on generation quality while using masked diffusion. We leverage LLaDA for query expansion, requiring non-trivial adaptation of its native generation code.

3 Methodology

3.1 Problem Formulation

Given a query q and corpus $\mathcal{D} = \{d_1, ..., d_n\}$, traditional dense retrieval computes:

$$\text{retrieve}(q) = \arg\max_{d \in \mathcal{D}} \text{sim}(E(q), E(d)) \tag{1}$$

where E is an embedding model. QUID modifies this to:

$$\text{retrieve}(q) = \arg\max_{d \in \mathcal{D}} \text{sim}(E(\text{expand}(q)), E(d)) \tag{2}$$

3.2 Diffusion-Based Expansion

QUID uses LLaDA's masked diffusion with the following process:

Key differences from autoregressive expansion:

- Block-based generation: Process in blocks, not left-to-right
- Confidence-based unmasking: Select tokens by prediction confidence
- Gumbel noise: Adds controlled stochasticity for diverse expansions

3.3 Implementation Details

- Embedding: BAAI/bge-m3 (1024 dimensions)
- Diffusion: GSAI-ML/LLaDA-8B-Instruct (bfloat16)
- Config: 32 steps, 32 tokens, temperature=0.5
- Compute: NVIDIA A10G GPU (24GB), Modal.com

4 Experiments

4.1 Evaluation Setup

We evaluate on four BEIR benchmark datasets with human relevance judgments:

Dataset	Domain	Queries	Docs
SciFact	Scientific claims	300	5,183
NFCorpus	Medical/Nutrition	323	3,633
FiQA	Finance Q&A	648	57,638†
TREC-COVID	COVID-19	50	171,332†

Table 1: BEIR datasets. †Sampled due to compute.

Methods: Vanilla (raw query), Template (static expansion), QUID (32-step diffusion), HyDE (GPT-3.5-turbo), BM25, BM25+RM3.

Metrics: nDCG@10, MRR@10, with bootstrap 95% CIs ($n = 1000$).

4.2 Main Results

Key Finding: Domain Determines Winner. QUID significantly outperforms HyDE on NFCorpus (+5.6%, $p < 0.05$) and shows advantages on FiQA and TREC-COVID. HyDE significantly outperforms QUID on SciFact (-6.8%, $p < 0.01$).

4.3 Efficiency Analysis

QUID is 1.3× slower than HyDE but runs locally with no API costs and full data privacy.

4.4 Ablation: Diffusion Steps Matter

This finding has important implications: the latency cost is intrinsic to diffusion's mechanism. Efficiency improvements require consistency distillation [11] or speculative decoding.

4.5 The "Editor vs Writer" Analysis

Surprising Finding: HyDE generates far more corpus-relevant vocabulary, yet QUID achieves better results on specialized domains. This suggests QUID's advantage comes from embedding space regularization—the diffusion process produces embeddings in a region better aligned with domain-specific documents—rather than vocabulary expansion per se.

4.6 Negative Results

Confidence Gating Fails. We attempted to skip diffusion for high-confidence queries (where expansion might hurt). However, LLaDA's confidence scores are near-zero ($\mu = 0.0001$) because masked diffusion models don't produce meaningful next-token probabilities like autoregressive models. Alternative gating mechanisms (embedding uncertainty, domain classification) remain future work.

Hybrid BM25+Dense Fails. Combining BM25 with QUID-expanded dense retrieval via score fusion ($\alpha = 0.5$) actually hurt performance (-11.3% vs QUID alone). This challenges the assumption that "combining signals always helps"—on NFCorpus, pure dense retrieval outperforms hybrid approaches.

Method	SciFact	NFCorpus	FiQA	TREC-COVID	Best Domain
Vanilla	0.5633	0.2793	0.4639	0.6259	—
Template	0.5512 (-2.2%)	0.2236 (-20.0%)*	0.4459 (-3.9%)*	0.6456 (+3.1%)	—
QUID	0.5618 (-0.3%)	0.3002 (+7.5%)*	0.4942 (+6.5%)*	0.6564 (+4.9%)	Medical, Finance
HyDE	0.6030 (+7.0%)*	0.2844 (+1.8%)	0.4857 (+4.7%)*	0.6362 (+1.6%)	Scientific

Table 2: Main results (nDCG@10). * = $p < 0.05$ vs vanilla. QUID wins on specialized domains; HyDE wins on scientific.

Corpus Type	Best Method	Margin
Medical (NFCorpus)	QUID	+7.5%***
Finance (FiQA)	QUID	+6.5%***
Biomedical (TREC-COVID)	QUID	+4.9% (n.s.)
Scientific (SciFact)	HyDE	+7.0%**

Table 3: Domain-dependent effectiveness. *** $p < 0.001$, ** $p < 0.01$

RM3 Also Fails. Classical pseudo-relevance feedback (BM25+RM3) hurts by -1.2% vs BM25, consistent with literature on specialized domains. This demonstrates that QUID's gains come from a fundamentally different mechanism than traditional expansion.

4.7 Extended Analysis: Probing QUID's Limitations

We conducted additional experiments to probe the boundaries of QUID's effectiveness. These reveal important limitations.

Recall@K Analysis. We measured Recall@K at multiple cutoffs to understand whether QUID finds relevant documents that vanilla misses.

Surprising Finding: QUID shows worse recall than vanilla at all cutoffs (-8.0% at R@10, -10.6% at R@50, -6.6% at R@100, -2.5% at R@200). This suggests QUID's nDCG@10 gains come from reranking relevant documents higher, not from finding documents that would otherwise be missed. HyDE outperforms both at higher K values.

Encoder Comparison. We tested whether QUID helps weaker encoders more than strong ones.

Surprising Finding: On this evaluation subset, QUID hurts all encoders tested, with the strongest encoder (BGE-M3) showing the largest degradation. This contradicts the hypothesis that diffusion expansion would help weaker encoders more.

Diversity Analysis. We measured whether QUID retrieves more semantically diverse documents.

Finding: QUID shows negligible diversity improvement (+0.0026 ILAD). On average, QUID finds only 1 additional relevant document that vanilla misses, while the two methods share 55% of their results.

Reconciling Positive and Negative Results. The discrepancy between our main results (Table 2) showing QUID gains and these extended analyses showing degradation likely stems from:

1. Query subset sensitivity: Performance varies significantly across query subsets
2. Metric sensitivity: nDCG@10 rewards ranking quality; Recall measures coverage
3. Evaluation variance: Small sample sizes (100 queries) have high variance

Method	Latency	Hardware	Bound
Vanilla	∼6ms	A10G GPU	Compute
Template	∼10ms	CPU	Compute
HyDE	∼1,200ms	OpenAI API	Network
QUID	∼1,600ms	A10G GPU	Compute

Table 4: Efficiency comparison. QUID is compute-bound (deterministic, batchable); HyDE is network-bound (variable, rate-limited).

Steps	nDCG@10	vs Vanilla	Latency
4	0.3104	-3.5%	214ms
8	0.3129	-2.7%	419ms
16	0.3167	-1.5%	898ms
32	0.3424	+6.4%	1,778ms

Table 5: Ablation on NFCorpus (100 queries). Fewer steps hurt—the iterative refinement is essential.

These findings suggest QUID's effectiveness is query-dependent rather than universal, and practitioners should validate on their specific query distribution before deployment.

5 Agentic Query Router (Month 4 Extension)

The main BEIR results show that always expanding is the wrong default: QUID helps when the bottleneck is a vocabulary gap (medical/finance) and can hurt on science-claim retrieval (SciFact). Review feedback asked for an agentic AI angle. We therefore treat expansion methods as tools and are designing an observe → decide → act controller on top of QUID—without replacing the diffusion contribution.

5.1 Controller Design

Loop: Query → plan tool ∈ {vanilla, QUID, HyDE} → expand → retrieve → critique confidence → optional retry → accept

Signals: token length, medical/finance/science cues, question form, unsupervised max corpus similarity

Tools: vanilla (skip), quid (LLaDA masked diffusion), hyde (hypothetical-document expand), critique, retry

Figure 3: Agentic retrieval controller (design target). Diffusion remains the expander; agency decides whether and when to call it.

We compare four policies on matched query slices:

1. always_vanilla — no expansion
2. always_QUID — blind diffusion expansion every time
3. single-shot agent — one routed tool call
4. multi-step agent — expand → critique → optional retry with a different tool

An oracle router (per-query best of {vanilla, QUID, HyDE} by labeled nDCG@10) provides an upper bound.

Metric	QUID	HyDE
Corpus vocab overlap	20.0%	97.1%
Novel terms introduced	1.5	47.7
Factual claim markers	0.1	3.5
NFCorpus nDCG@10	0.3002	0.2844

Table 6: Qualitative analysis. HyDE introduces more vocabulary yet QUID wins on NFCorpus—suggesting embedding regularization rather than vocabulary expansion.

Method	R@10	R@50	R@100	R@200
Vanilla	0.148	0.233	0.283	0.339
QUID	0.136	0.208	0.264	0.330
HyDE	0.136	0.232	0.295	0.362

Table 7: Recall@K on NFCorpus (100 queries). QUID underperforms vanilla at all K values.

5.2 Early Pilot Benchmarks

Pilots ran on Modal L4 (24GB) with LLaDA-8B-Instruct, 16 diffusion steps, gen_length 32, BGE-M3 embeddings, and an LLaDA-backed HyDE-style document prompt (OpenAI quota unavailable). These are pilot-scale (n=50 per dataset) signals, not frozen claims; larger confirmation runs are underway.

SciFact (science claims). Blind always-QUID hurts (−3.0 pts). Single-shot routing nearly recovers. Multi-step tool use reaches 0.6625 (+3.3% vs. vanilla, +6.3% vs. always-QUID, +3.5% vs. single-shot). When critique triggers a retry (6% of queries), nDCG@10 improves on 67% of those retries. Average expand tool calls stay near 1.06/query. Multi-step route mix: vanilla 42, HyDE 5, QUID 3 — the agent mostly skips expansion on this slice.

NFCorpus (medical). Always-QUID is brittle under this 16-step / 50-query setting (0.2704). Agentic routing recovers to 0.3053 (+3.5 pts vs. always-QUID) while remaining below vanilla and below oracle (0.3577), showing headroom for a learned router. Route mix: vanilla 27, QUID 22, HyDE 1.

Working takeaway. The research bet is not only "better prompting": selective tool-calling can improve retrieval when always-expand fails. We deliberately do not freeze agentic numbers as final claims yet. Month 4 next steps: fuller BEIR splits, stronger critique, learned routing toward oracle labels, and preprint polish.

6 Discussion

6.1 Why Does Diffusion Help Specialized Domains?

Our analysis suggests three complementary mechanisms:

1. Embedding Space Regularization: Diffusion's iterative unmasking acts as implicit regularization, producing embeddings in a region better aligned with specialized documents.

2. Noise as Feature: The Gumbel noise prevents overfitting to surface-level query terms, enabling matching to documents with different terminology but similar meaning.

3. Conservative Precision: QUID's minimal expansions avoid hallucination. In medicine and finance, a wrong fact is worse than no expansion.

Encoder	Vanilla	QUID	Change
BGE-M3 (SOTA)	0.3593	0.3241	−9.8%
MiniLM-L6 (Light)	0.3537	0.3225	−8.8%
MPNet-base	0.3813	0.3531	−7.4%

Table 8: Encoder comparison on NFCorpus (100 queries). QUID hurts all encoders tested.

Metric	Vanilla	QUID
ILAD (Intra-List Avg Distance)	0.4555	0.4581
Result Set Overlap	55.2%	
Unique Relevant Docs	—	1.0

Table 9: Diversity metrics on NFCorpus. ILAD difference is negligible (+0.0026).

6.2 When to Use What

QUID runs locally (no API costs, data privacy); HyDE requires API calls but wins on scientific corpora.

6.3 Limitations

- Latency: 1.6s/query is prohibitive for real-time search
- Single model: Only tested LLaDA-8B; other diffusion models may differ
- Corpus sampling: FiQA and TREC-COVID required sampling
- Query-dependent: Extended analysis shows QUID hurts on some query subsets; effectiveness varies significantly
- Recall vs. Ranking: QUID improves nDCG@10 but not recall—it reranks rather than discovers
- Encoder interaction: QUID may hurt all encoder types on certain evaluation subsets

7 Conclusion

We presented QUID, a diffusion-based query expansion method that treats queries as "noisy" document representations. Through rigorous BEIR evaluation, we established:

1. Domain-dependent effectiveness: QUID shows gains on medical (+7.5%) and finance (+6.5%) in our main evaluation; HyDE wins on scientific (+7.0%)
2. The "Editor" mechanism: Less vocabulary, potentially better results on specialized domains—suggesting embedding regularization rather than expansion
3. Mixed results: Extended analysis reveals QUID hurts recall at all cutoffs and may degrade performance on certain query subsets. This suggests QUID's effectiveness is query-dependent, not universal
4. Agentic extension (in progress): Early multi-step router pilots show selective tool use recovering from always-QUID failures and beating vanilla on SciFact (n=50); confirmation and a learned router are next

Method	NFCorpus	Δ	SciFact	Δ
always_vanilla	0.3305	—	0.6299	—
always_QUID	0.2704	−0.060	0.5999	−0.030
always_HyDE (LLaDA)	0.2677	−0.063	0.5550	−0.075
single-shot agent	0.3053	−0.025	0.6280	−0.002
multi-step agent	0.3053	−0.025	0.6625	+0.033
oracle router	0.3577	+0.027	0.7004	+0.071

Table 10: Agentic pilots (nDCG@10, n=50). Multi-step recovers vs. always-QUID on both slices; on SciFact it also beats vanilla.

If your corpus is...	Use	Expected Gain
Medical/health	QUID	+7–8%
Finance/business	QUID	+5–7%
Scientific/factual	HyDE	+7%
Mixed/unknown	Test both	varies

Table 11: Practitioner recommendations.

5. Honest limitations: Confidence gating, hybrid retrieval, and weaker encoder combinations all fail to improve over baselines

QUID reframes query expansion from "adding words" to "refining semantic representation"—an "Editor" to HyDE's "Creative Writer." The Month 4 research question is whether agentic tool use can decide when diffusion expansion helps—and recover when it does not. Practitioners should validate on their specific query distributions before deployment.

Reproducibility. Code available at: https://github.com/Zhreyu/quid. This document is a working draft; formal preprint coming soon.

References

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In NeurIPS.

[2] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216.

[3] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. arXiv preprint arXiv:2212.10496.

[4] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In EMNLP.

[5] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In SIGIR.

[6] Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In ICML.

[7] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jing Ou, Jun Hu, Jun Zhou, Yanyang Lin, Ji-Rong Wu, and Yang Yuan. 2025. Large language diffusion models. arXiv preprint arXiv:2502.09992.

[8] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In The SMART Retrieval System.

[9] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Notebook. In TREC.

[10] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marquez, Evan Pavlick, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In NeurIPS.

[11] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. 2023. Consistency models. In ICML.

[12] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. arXiv preprint arXiv:2303.07678.