Beyond Prompting: An Efficient Embedding Framework for Open-Domain Question Answering

Anonymous ACL submission

Abstract

Large language models (LLMs) have recently pushed open-domain question answering (ODQA) to new frontiers. However, prevailing retriever–reader pipelines often depend on multiple rounds of prompt-level instructions, leading to high computational overhead, instability, and suboptimal retrieval coverage. In this paper, we propose EmbQA, an embedding-level framework that alleviates these shortcomings by enhancing both the retriever and the reader. Specifically, we refine query representations via lightweight linear layers under an unsupervised contrastive learning objective, thereby reordering retrieved passages to highlight those most likely to contain correct answers. Additionally, we introduce an exploratory embedding that broadens the model's latent semantic space to diversify candidate generation and employs an entropy-based selection mechanism to choose the most confident answer automatically. Extensive experiments across three open-source LLMs, three retrieval methods, and four ODQA benchmarks demonstrate that EmbQA substantially outperforms recent baselines in both accuracy and efficiency.

1 Introduction

Recent advances in large language models (LLMs) (Achiam et al., 2023; Dubey et al., 2024) have propelled Open-Domain Question Answering (ODQA) to new heights. A central strategy in ODQA involves retrieving relevant knowledge (Lei et al., 2023) and then integrating it with LLMs acting as readers to synthesize accurate answers. This retriever-reader approach has shown promise in overcoming the inherent limitations of LLMs (Mialon et al., 2023).

Yet, prevailing retriever-reader architectures face two key limitations. First, retrievers (Karpukhin et al., 2020; Lei et al., 2023) yield abundant candidate passages, they fail to effectively prioritize those containing definitive answers. This is evidenced by their low ground truth recall in top-ranked results, where directly retraining retrievers or applying prompt-level reranking (Chuang et al., 2023) proves impractical due to prohibitive computational costs (Zhuang et al., 2024a) and inherited inefficiency from multi-turn processes. Second, while reader relies on multi-turn prompt-level strategies such as self-verification (Gao et al., 2023), or additional summarization (Kim et al., 2024), which requires expensively inference cost by LLM, leads to computational inefficiency and instability in the answer selection.

To address these limitations, we propose **Emb**edding-Driven Reranking and Answer Generation Framework for Open Domain **QA** Driven (**EmbQA**), which utilizes the embedding strategy to enhance both efficiency and accuracy in both retriever and reader.

In retriever, we propose an embedding-level rerank framework that leverages candidate answers generated by LLMs to guide query refinement via unsupervised contrastive learning. Compared with existing LLM prompting-based reranking frameworks (Karpukhin et al., 2020; Lei et al., 2023) which only focus few candidate passages due to the high computational cost of language inference, in our proposed method, we are able to fully explore the whole selected knowledge based on a learnable embedding layer. By mapping both queries and candidate sentences into the retrieval space and refining the query embedding with only a simple linear combination, our approach effectively reranks retrieved passages to prioritize those most likely to contain the correct answer.

In reader, as many existing works suggest that there is a latent presence of prerequisite knowledge within the model's parameter space (Ye et al., 2025), and inserting a single compressed token can activate the neural pathways in LLM to generate the correct answer (Cheng et al., 2024). Building on this, we propose an order-statistic-based measure for exploratory embedding generation. This method allows us to explore statistically orthogonal directions by inserting only one token-sized embedding. It not only enhances diversity but also improves efficiency, as it eliminates the need for additional prompting rounds for summarization or verification. By utilizing perturbed predictive entropy, we can filter out uncertain answer candidates.

In summary, our contributions are summarized as follows:

1. We develop an embedding-level rerank framework that leverages candidate answer guidance via unsupervised contrastive learning to optimize retrieval effectiveness.
2. We propose an order-statistic-based single-token embedding strategy that activates latent knowledge within LLMs, reduces multi-turn prompting overhead and diversifies candidate generation.
3. Extensive experiments on three state-of-the-art open-sourced LLMs, and three retrieval methods across four ODQA datasets, demonstrate that our framework significantly outperforms existing prompt-level frameworks in both efficiency and accuracy.

2 Related Work

Rerank of Retriever in Open-Domain QA. Open-domain question answering (ODQA) (Voorhees et al., 1999) typically adopts a retriever-reader framework (Chen et al., 2017), where a retriever selects relevant documents from extensive corpora and a reader synthesizes them into answers. Retrieval techniques generally fall into two categories: lexical methods (e.g., BM25 (Robertson et al., 2009)) and dense models leveraging sentence embeddings (e.g., DPR (Karpukhin et al., 2020) and Contriever (Lei et al., 2023)). However, existing ODQA systems often underutilize retrieval capabilities, as top-ranked documents frequently lack comprehensive answer coverage (Zhuang et al., 2024a). Reranking strategies can mitigate this issue by prioritizing critical documents, thereby improving answer coverage and accuracy (Zhuang et al., 2024b). In the era of large language models, these strategies are mostly implemented at the prompt level (Meng et al., 2024; Li et al., 2024), yet such methods have been found inefficient, time-consuming (Zhuang et al., 2024b), and unstable (Wu et al., 2024b). While emerging work has begun to explore word-level reranking, this area remains underexplored. To address these challenges, we propose an embedding-level mechanism that integrates reranking without requiring labelled data, offering a more efficient and stable alternative to prompt-level methods.

Prompt-level Framework of Reader in Open-Domain QA. Prompt-level framework of Reader enhances language models by comparing multiple candidate solutions, either by selecting the best option (Kim et al., 2024) or synthesizing several outputs for final prediction (Zhang et al., 2024) and self-verification (Gao et al., 2023). In ODQA, prevailing paradigms generate and evaluate candidate answers through summaries of retrieved content (Giorgi et al., 2023; Gao et al., 2023; Kim et al., 2024). However, these frameworks rely mainly on prompt-level mechanisms, which require multiple rounds of prompting, thus incurring inefficiencies and sensitivity to answer quality. Recent work has begun exploring word-level framework via key term masking for self-correction (Wu et al., 2024c), yet this line of inquiry remains nascent. In contrast, our approach advances to the embedding level by incorporating diversity information during candidate generation to steer the model toward candidate sets more likely to contain the correct answer, and an entropy-based candidate filtering mechanism further ensures a more efficient selection process than prompt-level methods.

3 Methodology

3.1 Overview and Problem Description

Overview. We propose EmbQA, a two-stage framework that addresses the aforementioned limitations in open-domain QA. As illustrated in Figure 1, EmbQA consists of: (1) *Retriever* that refines the query through unsupervised contrastive learning, allowing it to effectively re-rank passages so that those potentially containing correct answers are prioritized; and (2) *Reader* that broadens the semantic space for answer generation by injecting a lightweight exploratory embedding derived from a normal distribution. This exploratory embedding nudges the model to discover a more diverse set of potential answers. Finally, we rely on an entropy-based criterion over the model's output logits to select the best answer without resorting to multiple rounds of prompts.

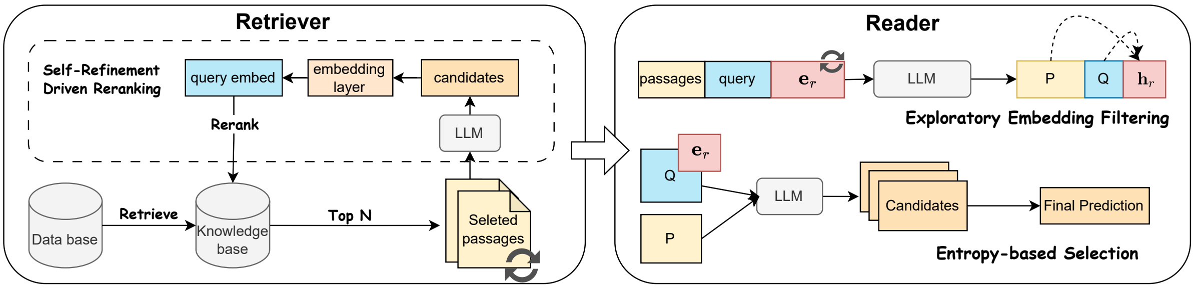

Figure 1: Overview of the EmbQA framework. *Retriever* module constructs a knowledge base by retrieving passages from a large corpus and then refines the query via an embedding layer under unsupervised contrastive learning to prioritize passages rich in answer-critical cues. Then *Reader* module integrates an exploratory embedding into the query to diversify candidate generation and employs an entropy-based selection mechanism to pick the final answer with the lowest uncertainty, ultimately enhancing both efficiency and overall performance in ODQA.

Problem Description. Open-domain question answering (ODQA) typically relies on external knowledge beyond a single context. Following the retriever–reader pipeline (Chen et al., 2017; Lee et al., 2019), we first form a *knowledge base* $\mathcal{B}$ by retrieving relevant passages from the large corpus $\mathcal{D}$ given a query q, then select the top-N passages $\mathcal{C}_N$ from $\mathcal{B}$ for candidate answer generation and subsequently refine the query representation $\mathbf{e}_{q_{\text{new}}}$ to re-rank $\mathcal{B}$, emphasizing passages more likely to contain the correct answer. Then, we introduce a lightweight *exploratory embedding* $\mathbf{e}_r$ to diversify the candidate answers. Finally, we compute logit-based entropy to select the best candidate as the final prediction.

3.2 Retriever: Self-Refinement Driven Reranking

Despite significant progress, existing retrieval frameworks still struggle to effectively prioritize passages containing definitive answers. To address this, we propose a reranking framework that increases the likelihood of including ground-truth passages, thereby improving retrieval quality. Additionally, prior prompt-level reranking methods have proven inefficient and time-consuming (Zhuang et al., 2024b). In contrast, EmbQA adopts an embedding-based reranking strategy that is both more scalable and more efficient than previous prompt-level approaches (Li et al., 2024; Zhuang et al., 2024a).

Candidate Sentence Generation. Given a query and its retrieved passages, we use a specialized prompt to generate K candidate answers $y = [y_1, \ldots, y_K]$ via an LLM. While prior studies (Lazaridou et al., 2022; Weng et al., 2023) employ stochastic decoding to enhance diversity, we adopt an approach inspired by (Kim et al., 2024), explicitly prompting the LLM to generate K answer candidates. Following the empirical findings of (Kim et al., 2024), which indicate no performance gains with larger K, we set $K = 2$ in this research.

Rerank by Unsupervised Contrastive Learning. Directly tuning a large retriever or relying on multiple rounds of prompt-based reranking can be computationally prohibitive. To address this, we propose a lightweight method that *refines* the query representation, enabling more precise passage selection without retriever-wide fine-tuning.

Concretely, let a frozen retriever map the original query q and an LLM-generated candidate answers y into a shared representation space, producing embeddings $\mathbf{e}_q$ and $\mathbf{e}_y$. We then form a new query embedding $\mathbf{e}_{q_{\text{new}}}$ via a simple linear combination:

$$\mathbf{e}_{q_{\text{new}}} = \mathbf{W}_1\mathbf{e}_y + \mathbf{W}_2\mathbf{e}_q. \tag{1}$$

where W_1 and W_2 are the only trainable parameters. This design is far more efficient than modifying the entire retriever, while still capturing critical cues from the candidate sentence.

To learn $\mathbf{W}_1$ and $\mathbf{W}_2$, we adopt an unsupervised contrastive loss (Oord et al., 2018) that encourages $\mathbf{e}_{q_{\text{new}}}$ to focus on passages containing the candidate answers from y. Specifically, we treat passages that contain at least one candidate as *positive*, and those that do not as *negative*. Because negatives are significantly more abundant, we sample them at a fixed ratio of 5:1 relative to positives in each training batch to maintain balance. Once the parameters are updated, we use $\mathbf{e}_{q_{\text{new}}}$ to re-query the retriever, effectively *re-ranking* the knowledge base $\mathcal{B}$ so that passages with correct answers appear more prominently. This addresses the inefficiency of multi-turn prompt-level reranking and increases the likelihood that the subsequent Reader module receives high-quality evidence.

3.3 Reader: Enhancing Generation via Exploratory Embedding

Unlike existing ODQA patterns such as SuRe (Kim et al., 2024), which relies on summarization and prompt-level candidate selection strategies, EmbQA removes summarization and replaces prompt-level candidate selection with embedding-level entropy-based selection. Furthermore, our approach diversifies candidate generation via the exploratory embedding mechanism with variance gate filtering, which guides the model to explore a broader semantic space.

Exploratory Embedding Filtering We introduce an *exploratory embedding* to diversify candidate generation. Suppose the LLM $\mathcal{M}$ has an embedding dimension of D. We sample a random vector $\mathbf{e}_r \in \mathbb{R}^D$ from a standard normal distribution and concatenate it with the query q and retrieved context $\mathcal{C}_N$ at inference time. We then extract the hidden representation $\mathbf{h}_r$ corresponding to $\mathbf{e}_r$ from the penultimate layer of $\mathcal{M}$, following the practice of Liu et al. (2024) to capture sentence-level semantics.

Inspired by Jain et al. (2014), who shows that encouraging orthogonality among a set of vectors can be achieved by minimizing their maximum inner products, we adopt the inner product as a measure of diversity. Our goal is to find an $\mathbf{h}_r$ that is maximally *inconsistent* (i.e., yields the smallest inner product) with the concatenated representation $E(\mathcal{C}_N; q)$. However, directly optimizing over the entire embedding space is computationally expensive—especially when considering tokens that have not yet been decoded. To address this, we assume that token embeddings follow a Gaussian distribution and derive an analytical approximation using order statistics. Specifically, we sort the elements of $\mathbf{h}_r$ in descending order and define Δ_i as the gap between the i-th and $(i+1)$-th largest elements. We then show that minimizing the expected inner product between $\mathbf{h}_r$ and a set of Gaussian vectors is approximately equivalent to minimizing the squared sum of the top-p differences: $S_{\mathbf{e}_r} = \sum_{i=1}^{p} \Delta_{(i)}^2$. (A detailed theoretical discussion justifying this approximation in relation to Jain et al. (2014)'s claim is provided in Appendix B.) We repeat the sampling process until we obtain an $\mathbf{e}_r$ such that $S_{\mathbf{e}_r}$ falls below a preset threshold T.

Entropy-Based Selection. Once a suitable $\mathbf{e}_r$ is obtained, we regenerate candidate answers $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_K]$ using the LLM with the retrieved context $\mathcal{C}_N$, the original query q, and the selected exploratory embedding. Rather than relying on multi-turn prompts for candidate refinement, we leverage the LLM's own output uncertainties. Specifically, for each candidate answer, we compute a logit-based entropy score. Motivated by recent findings (Wu et al., 2024a; Wang et al., 2025) that lower entropy correlates with higher confidence, we select the candidate with the lowest entropy:

$$\hat{a} = \operatorname*{argmin}_{\hat{y} \in \{\hat{y}_1, \ldots, \hat{y}_K\}} \text{Entropy}(\hat{y}). \tag{2}$$

This embedding-level, entropy-based selection strategy eliminates the need for additional prompt rounds, making the answer generation process both efficient and robust.

4 Experiments

4.1 Setups

Evaluation Datasets. We evaluate EmbQA on zero-shot QA across four ODQA datasets: HotpotQA (Yang et al., 2018), 2WikiMulti-hopQA (2Wiki) (Ho et al., 2020), Natural Questions (NQ) (Kwiatkowski et al., 2019), and WebQuestions (WebQ) (Berant et al., 2013). For NQ and WebQ, we use their original test splits with the 21M English Wikipedia dump (Karpukhin et al., 2020) as the retrieval corpus. For all datasets, we adopt the implementation splits provided by Trivedi et al. (2023) and Kim et al. (2024), along with their respective document corpora.

Metrics. We use exact match (EM) and F1 score as evaluation metrics. Following Rajpurkar et al. (2016), we normalize predictions and gold answers by lowercasing and removing punctuation to ensure consistency.

Baselines. We compare EmbQA with the following methods: (1) *No Retrieval* generates answers applying an LLM in a closed-book setting without retrieved passages. (2) *Retrieval Only* appends retrieved passages to the input prompt. (3) *Chain-of-Thoughts* (Kojima et al., 2022; Wei et al., 2022) augments the prompt with zero-shot chain-of-thought reasoning. (4) *Self-Verification* (Weng et al., 2023) generates multiple answer candidates via random sampling and selects the most plausible one by verifying its reasoning using conditional masking. (5) *SuRe* (Kim et al., 2024) produces candidate

Method/Dataset	HotpotQA		2Wiki		NQ		WebQ		Average	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
No Retrieval	20.8	29.1	12.2	16.2	20.6	26.6	17.2	25.8	17.7	24.4
Retrieval Only	25.4	37.2	16.6	21.1	26.0	32.8	22.2	31.2	22.6	30.6
Chain-of-Thought	27.0	39.8	15.4	21.8	27.2	33.5	28.8	37.8	24.6	33.2
Self-Verification	32.8	49.5	21.0	23.5	28.0	37.7	27.2	40.2	27.4	38.0
SuRe	38.8	53.5	23.8	31.0	36.6	47.9	34.4	48.5	33.4	45.3
EmbQA (Ours)	**42.0**	**55.8**	**27.4**	**36.6**	**42.2**	**54.4**	**38.2**	**52.1**	**37.5**	**49.7**

Table 1: Comparison of prompt-level frameworks on four open-domain QA datasets (HotpotQA, 2Wiki, NQ, and WebQ) using LLaMA 3.1. All methods retrieve the top-10 relevant passages using BM25. The EmbQA framework outperforms existing prompt-level approaches across all datasets.

answers and selects the most plausible one by conditional summarization as the final prediction.

Implementation Details. Our framework requires modifications at the embedding level, which necessitates the use of open-sourced LLMs. We conduct experiments with three state-of-the-art models: LLaMA-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Yang et al., 2024). We set the decoding temperature to 0.0 to ensure greedy decoding, to eliminate the effect of random sampling (Sun et al., 2023). For retrieval, we employ three approaches: a lexical-based retriever (BM25) (Robertson et al., 2009) and two dense retrievers (DPR-multi (Karpukhin et al., 2020) and Contriever (Lei et al., 2023)). We use Elasticsearch for BM25 and the BEIR toolkit for DPR and Contriever,[1] respectively. In our framework, an initial retriever retrieves candidate passages, which are then reranked based on modifications at the embedding level. Notably, when BM25 is used as the initial retriever—owing to its lexical nature and inability to generate sentence embeddings—we employ Contriever for reranking; in contrast, when DPR or Contriever serves as the initial retriever, the same model is used throughout. We use consistent prompts across all datasets (Appendix A) and fix $K = 2$ in all experiments following Kim et al. (2024). Although iterative reranking is theoretically possible, we perform only a single reranking pass given the limited performance gains relative to the increased computational cost. In the exploratory embedding stage, a variance gating threshold of 0.05 is applied.

[1] https://www.elastic.co/, https://github.com/beir-cellar/beir

4.2 Main Results

EmbQA Outperforms Prompt-Level Methods. Table 1 presents the performance of lines of prompt-level frameworks on four open-domain QA datasets using LLaMA 3.1 with BM25-based retrieval of the top-10 passages. Notably, augmenting retrieved passages with prompting generally improves ODQA accuracy over a pure retrieval strategy. However, our proposed embedding-level framework consistently outperforms these prompt-level approaches across all datasets. For instance, on HotpotQA, our method achieves an Exact Match (EM) of 42.0 and an F1 score of 55.81, representing improvements of approximately 3.2 and 2.3 points over the best prompt-level baseline (SuRe). Similar gains are observed on the remaining datasets, underscoring the efficacy of leveraging embedding-level information to enhance LLM performance in open-domain QA tasks.

Robust Generality of EmbQA Across Setups. Table 2 further demonstrates the compatibility of our embedding-level framework across three LLMs and four question-answering datasets with three retrieval methods. Our framework consistently outperforms the retrieval-only and SuRe (Kim et al., 2024) baselines in nearly every setting. For instance, on LLaMA 3.1 with DPR, EmbQA achieves 29.8 EM and 36.34 F1 on HotpotQA, substantially exceeding BM25+SuRe's performance around 4.8 on EM and 4.46 on F1. Comparable improvements are observed across different retrievers and models, underscoring the generalizability and robustness of our approach in enhancing open-domain QA performance.

Model	Retriever & Framework	HotpotQA EM	HotpotQA F1	2Wiki EM	2Wiki F1	NQ EM	NQ F1	WebQ EM	WebQ F1
LLaMA3.1 8B-Ins	BM25	25.4	37.2	16.6	21.1	26.0	32.8	22.2	31.2
	+SuRe	38.8	53.5	23.8	31.0	36.6	47.9	34.4	48.5
	+EmbQA (ours)	**42.0**	**55.8**	**27.4**	**36.6**	**42.2**	**54.4**	**38.2**	**52.1**
	DPR	20.6	21.7	10.8	13.5	25.0	34.2	23.8	34.4
	+SuRe	25.0	31.9	14.2	16.0	38.8	52.3	36.0	49.6
	+EmbQA (ours)	**29.8**	**36.3**	**16.8**	**21.0**	**43.0**	**54.4**	**38.0**	**52.0**
	Contriever	22.6	35.4	16.6	20.7	25.8	32.8	25.2	34.2
	+SuRe	33.8	50.6	21.0	29.3	39.0	52.8	34.4	48.5
	+EmbQA (ours)	**36.6**	**52.7**	**26.4**	**34.2**	**42.2**	**53.6**	**36.0**	**49.6**
Mistral v0.2 7B-Ins	BM25	21.2	29.2	13.8	21.7	18.8	25.3	19.0	26.1
	+SuRe	32.2	**46.1**	17.8	30.1	35.2	45.1	31.6	45.7
	+EmbQA (ours)	**34.8**	44.3	**18.6**	**30.5**	**35.8**	**46.0**	**35.8**	**48.1**
	DPR	7.8	11.0	3.8	4.5	22.2	26.7	18.8	27.7
	+Sure	15.0	21.8	6.4	8.5	40.0	**51.8**	32.6	**47.7**
	+EmbQA (ours)	**16.2**	**23.3**	**7.6**	**9.6**	**40.2**	49.4	**33.4**	46.0
	Contriever	19.4	28.6	13.6	20.7	21.8	27.4	17.8	24.4
	+SuRe	28.0	41.6	17.2	25.4	39.8	51.6	30.2	**45.0**
	+EmbQA (ours)	**29.8**	**42.3**	**17.4**	**26.2**	**40.6**	**51.8**	**31.6**	43.0
Qwen 2.5 7B-Ins	BM25	28.6	37.1	20.2	24.1	24.0	29.4	22.6	31.4
	+Sure	43.6	54.7	28.4	**34.1**	41.6	49.0	36.6	47.3
	+EmbQA (ours)	**44.6**	**55.6**	**28.8**	33.8	**42.4**	**49.2**	**38.2**	**48.7**
	DPR	8.8	9.8	5.6	7.1	29.2	32.6	25.6	31.1
	+Sure	21.8	27.3	12.2	16.1	45.4	54.6	38.4	49.6
	+EmbQA (ours)	**22.6**	**29.1**	**13.8**	**17.3**	**45.8**	**54.7**	**38.6**	**50.1**
	Contriever	27.0	34.0	17.6	20.0	26.6	31.9	21.0	29.1
	+Sure	38.8	**50.3**	23.8	30.4	44.0	**52.9**	36.4	48.1
	+EmbQA (ours)	**39.0**	50.2	**24.4**	**30.9**	**45.2**	50.5	**37.0**	**48.6**

Table 2: Exact Match (EM %) and F1 score performance of LLaMA 3.1, Mistral v0.2, and Qwen 2.5 across HotpotQA, 2Wiki, NQ, and WebQ datasets. Each model is evaluated using three retrieval methods: BM25 (lexical retriever), DPR, and Contriever (dense retrievers). Results are reported for retrieval-only, SuRe, and our proposed framework. Across all models, retrievers, and datasets, our framework consistently outperforms both the SuRe baseline and retrieval-only approaches.

Efficiency. Table 3 compares the execution time and output token requirements of SuRe (Kim et al., 2024) and our proposed framework across four datasets using LLaMA 3.1 with Contriever retrieval. Our method consistently reduces both query time and output token requirement by a significant margin. For instance, on HotpotQA, our framework processes a query in 0.53 minutes and only requires approximately $0.99k$ output tokens to generate a final prediction, compared to 1.56 minutes and $3.51k$ tokens for SuRe (Kim et al., 2024). Similar efficiency gains are observed across 2Wiki, NQ, and WebQ datasets, underscoring the superior computational efficiency of our approach.

Dataset	Method	Time/query (min) ↓	Tokens /query ↓
HotpotQA	SuRe	1.56	$3.51k$
	EmbQA (ours)	**0.53**	**0.99k**
2Wiki	SuRe	1.57	$3.43k$
	EmbQA (ours)	**0.54**	**1.20k**
NQ	SuRe	1.43	$4.39k$
	EmbQA (ours)	**0.54**	**0.84k**
WebQ	SuRe	1.58	$3.91k$
	EmbQA (ours)	**0.56**	**1.31k**

Table 3: Comparison of execution time and output token requirement per query between SuRe and our proposed framework EmbQA across four open-domain QA datasets (HotpotQA, 2Wiki, NQ, and WebQ), using LLaMA 3.1 with Contriever retrieval. Our method significantly reduces query time and output token requirement in all datasets.

5 Analysis

Ablation Studies. We conducted an ablation study to assess the contribution of each component in our overall framework. We evaluated four configurations: (1) Retrieval Only, (2) EmbQA (our full framework), (3) EmbQA **w/o** exploratory embedding (without the exploratory embedding module), and (4) EmbQA **w/o** exploratory embedding & rerank (without both the exploratory embedding and the reranking module). Figure 2 reports the Exact Match (EM) and F1 scores on HotpotQA, 2Wiki, NQ, and WebQ. The results indicate that each component contributes additively to performance improvements. For example, on HotpotQA, the full EmbQA model achieves 36.6 EM and 52.7 F1, which is substantially higher than the 22.6 EM and 35.4 F1 obtained in the Retrieval Only setting. Removing the exploratory embedding module results in a performance drop to 35.2 EM and 50.9 F1, and further removing the reranking module degrades the scores to 33.6 EM and 50.2 F1. Similar trends across all datasets demonstrate a sequential degradation in performance as each module is removed, highlighting the additive contributions of each component in our framework.

Why Prompt-Level Rerank Framework Fail in Existing ODQA Framework? Existing ODQA systems (Kim et al., 2024) have demonstrated that prompt-level reranking can be ineffective or even detrimental to performance. We posit that this failure stems from the inability of prompt-level reranking to reliably elevate ground-truth passages among the top-10 retrieved results, and in some cases, it may even reduce their presence. Table 4 presents a

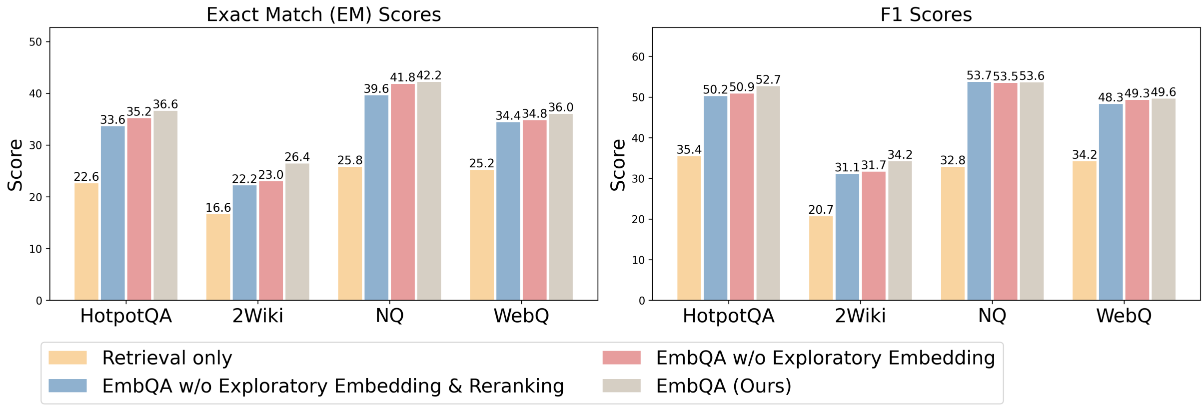

Figure 2: Ablation study on HotpotQA, 2Wiki, NQ, and WebQ datasets using LLaMA 3.1 with Contriever retrieval. The results compare four settings: (1) Retrieval Only, (2) Full EmbQA, (3) EmbQA without the exploratory embedding module, and (4) EmbQA without both the exploratory embedding and reranking modules. Each component contributes crucially to the overall performance, as evidenced by incremental improvements in Exact Match (EM) and F1 scores.

Retriever& Rerank Framework	**HotpotQA**		**2Wiki**		**NQ**		**WebQ**	
	Avg. GT @Top-10	**Time /Query(s)**	**Avg. GT @Top-10**	**Time /Query(s)**	**Avg. GT @Top-10**	**Time /Query(s)**	**Avg. GT @Top-10**	**Time /Query(s)**
BM25	1.16	–	0.81	–	1.50	–	1.88	–
+Prompt Level	1.06	12.52	1.09	12.62	1.62	12.65	2.70	12.69
+Embedding Level (Ours)	**1.42**	**1.33**	**1.21**	**1.54**	**2.57**	**1.90**	**4.18**	**2.31**
DPR	0.28	–	0.30	–	1.79	–	3.04	–
+Prompt Level	0.64	13.23	0.34	12.52	1.75	12.66	3.68	12.63
+Embedding Level (Ours)	**1.01**	**2.42**	**1.13**	**1.27**	**2.41**	**2.00**	**4.25**	**2.22**
Contriever	1.47	–	0.99	–	1.98	–	2.87	–
+Prompt Level	1.37	12.54	1.36	12.95	2.01	13.16	3.02	13.05
+Embedding Level (Ours)	**1.87**	**1.12**	**1.49**	**2.93**	**2.55**	**2.12**	**4.31**	**2.69**

Table 4: Retrieval analysis on HotpotQA, 2Wiki, NQ, and WebQ datasets using LLaMA3.1. For each dataset, the two metrics are: **Average Ground Truth Passages in Top-10** (the average number of ground-truth passages present among the top-10 retrieved results) and **Time Consumption Per Query (second)** (the time taken for processing each query in seconds). Higher values in the first metric indicate that our reranking framework surfaces relevant passages more effectively compared to the BM25 baseline.

comparison between our embedding-level rerank framework and a state-of-the-art prompt-level approach (Zhuang et al., 2024a) across four datasets (HotpotQA, 2Wiki, NQ, and WebQ) using three different retrievers (BM25, DPR, and Contriever). Two key metrics are reported: (i) the average number of ground-truth passages in the top-10 results, and (ii) the per-query processing time (in seconds). Existing prompt-level rerank frameworks often struggle to effectively enhance this metric, and in some cases, they even weaken it—resulting in extremely unstable downstream ODQA performance. For example, on the HotpotQA dataset with BM25, the prompt-level method reduces the average ground-truth count from 1.16 to 1.06, potentially impairing downstream ODQA performance. In contrast, our embedding-level approach increases this count from 1.16 to 1.42. Similar trends are observed with DPR and Contriever, underscoring the broad applicability of our approach. Moreover, our experiments reveal that when substituting our embedding-level framework with a prompt-level alternative within the overall system, performance degrades significantly (see Appendix C.1). In terms of efficiency, our embedding-level framework dramatically reduces query processing time. For instance, with BM25 on HotpotQA, the prompt-level rerank requires 12.52 seconds per query, whereas our method reduces this to 1.33 seconds. These experimental results across both metrics clearly demonstrate that our embedding-level rerank framework not only more effectively promotes relevant passages but also substantially enhances system efficiency.

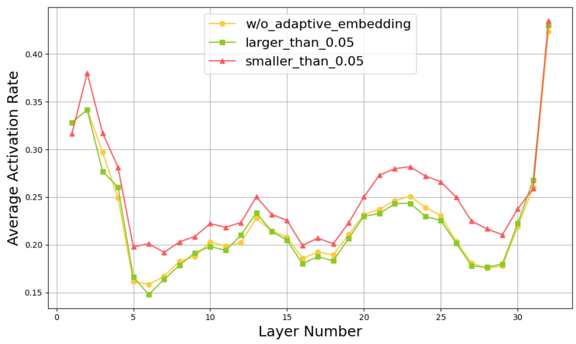

Figure 3: MLP layer activation rates across Transformer layers for the first five tokens of generated answers, sampled 200 times. The curves compare our framework without exploratory embedding and with exploratory embedding with variance gating larger than 0.05 or smaller than 0.05.

Dataset	Method	BM25	DPR	Contriever
		Candidates Coverage (%): ↑		
HotpotQA	EmbQA	**58.0**	**50.0**	**51.0**
	w/o exploratory	55.0	46.0	48.2
2Wiki	EmbQA	**51.0**	**43.6**	**31.6**
	w/o exploratory	47.0	40.2	28.2
NQ	EmbQA	**46.6**	**43.6**	**45.8**
	w/o exploratory	43.0	41.8	41.2
WebQ	EmbQA	**57.8**	**56.6**	**51.8**
	w/o exploratory	54.6	52.0	49.2

Table 5: Candidates coverage analysis on four datasets (HotpotQA, 2Wiki, NQ, and WebQ) across three retrievers (BM25, DPR, and Contriever). The metric, **Candidates coverage (%)**, represents the proportion of candidates that contain the ground truth answer. We compare our framework EmbQA with exploratory embedding injection against the variant without it (- Exploratory Embedding). The results demonstrate that incorporating exploratory embedding injection enhances diversity and increases the likelihood of covering the correct answer.

Effect of Exploration with Exploratory Embedding. Although we have theoretically demonstrated that minimizing the variance allows us to filter exploratory embedding that deviate from the context and query direction B, this conclusion may not be entirely intuitive. A natural question arises: why choose 0.05 as the variance gating threshold? Following Naik et al. (2024), we investigate this from the perspective of neuron activations in the Transformer's MLP layers. Figure 3 illustrates the activation rates across Transformer layers for the first five tokens of generated answers, sampled 200 times. We observe that when exploratory embedding is applied with a variance gate lower than 0.05, each layer—particularly from the 5th to the 30th, exhibits an increase in activation rates of roughly 3–4%. However, when the variance exceeds 0.05, the activation rates remain nearly unchanged or even slightly lower compared to the setting without exploratory embedding. This phenomenon suggests that our exploratory embedding with the variance gate stochastically triggers more diverse neural pathways.

At the token level, we further explore whether exploratory embedding enhances the possibility of generating the correct candidates. Table 5 presents the candidate sentence coverage, defined as the percentage of generated candidate sentences that contain the ground truth answer, across four datasets and three retrievers. We compare our full framework EmbQA, which incorporates exploratory embedding injection, with a variant that excludes this operation (**w/o** Exploratory). The results consistently show that exploratory embedding improves coverage across all settings. For example, on HotpotQA with BM25, coverage increases from 55.0% without exploratory embedding to 58.0% with it, with similar improvements observed for DPR, Contriever, and across the other datasets. These findings collectively indicate that exploratory embedding injection not only promotes more diverse neural activation but also increases the likelihood of including the correct answer in the generated candidates.

6 Conclusion

We introduce EmbQA, an embedding-level framework for open-domain QA that improves efficiency over multi-turn prompt-based systems. By refining query representations with lightweight linear layers trained via unsupervised contrastive learning, our approach reorders retrieved passages to prioritize those most likely to contain correct answers. Additionally, an exploratory embedding with an entropy-based selection mechanism enhances candidate diversity and streamlines self-verification. Experiments across multiple ODQA benchmarks, retrieval methods, and state-of-the-art LLMs show that EmbQA consistently outperforms prompt-level approaches in accuracy and efficiency.

Limitations

Although EmbQA significantly enhances both efficiency and accuracy in open-domain QA, it comes with several limitations. First, our approach relies on access to an open-source LLM to modify embeddings at the model level, which may not be feasible for scenarios where only black-box API-based models are available. This constraint limits the direct applicability of our method to widely used proprietary models such as GPT-4 or Claude. Second, while EmbQA reduces the computational overhead associated with multi-turn prompt-based methods, it introduces an additional embedding-level training step. Although this step is lightweight compared to full retriever fine-tuning, it still requires additional computational resources, which may not be ideal for resource-constrained environments. Lastly, our framework assumes that reranking retrieved passages based on learned query refinements will consistently improve answer selection. However, its effectiveness depends on the quality of the retrieved passages—if the initial retrieval fails to include informative passages, reranking alone may not be sufficient to bridge the gap. Future work can explore adaptive retrieval mechanisms to further enhance robustness across diverse retrieval conditions.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

Stéphane Boucheron and Maud Thomas. 2012. Concentration inequalities for order statistics. *Electronic Communications in Probability*, 17(none).

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *Preprint*, arXiv:2405.13792.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147, Toronto, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2024. A mathematical perspective on transformers. *Preprint*, arXiv:2312.10794.

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Wang, and Arman Cohan. 2023. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8177–8199, Singapore. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yajit Jain, Deepak Narayanan, and Leon Zhang. 2014. Almost orthogonal vectors.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain QA of LLMs. In *The Twelfth International Conference on Learning Representations*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024. Self-prompting large language models for zero-shot open-domain QA. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 296–310, Mexico City, Mexico. Association for Computational Linguistics.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. 2024. Fantastic semantics and where to find them: Investigating which layers of generative LLMs reflect lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14551–14558, Bangkok, Thailand. Association for Computational Linguistics.

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Ranked list truncation for large language model-based re-ranking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 141–151.

Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*. Survey Certification.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. 2024. DIVERSITY OF THOUGHT IMPROVES REASONING ABILITIES OF LARGE LANGUAGE MODELS.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

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.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Keheng Wang, Feiyu Duan, Peiguang Li, Sirui Wang, and Xunliang Cai. 2025. LLMs know what they need: Leveraging a missing information guided framework to empower retrieval-augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2379–2400, Abu Dhabi, UAE. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. 2024a. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.

Xuyang Wu, Zhiyuan Peng, Krishna Sravanthi Rajanala Sai, Hsin-Tai Wu, and Yi Fang. 2024b. Passage-specific prompt tuning for passage reranking in question answering with large language models. In *The Second Workshop on Generative Information Retrieval*.

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024c. Large language models can self-correct with key condition verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12846–12867, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3602–3622, Bangkok, Thailand. Association for Computational Linguistics.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024a. Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 358–370, Mexico City, Mexico. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024b. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.

A Prompt Design

In this section, we present the specific prompts used for the experiments in Section 4.1.

A.1 Answer Candidates Generation

In Listing 1, we present the prompt p_{can} which is used to generate K answer candidates from the given question and N retrieved passages. Here, we present the case of $K = 2$.

Prompt for Answer Candidates Generation

Passage #1 Title: {Passage #1 Title}
Passage #1 Text: {Passage #1 Text}

...

Passage #N Title: {Passage #N Title}
Passage #N Text: {Passage #N Text}

Question: {Question}

Answer:
```
text = f"Below are {n_articles} passages related to the question at the end.
After reading the passages, provide two correct candidates for the answer to the question.
Each answer should be in the form: (a) xx, (b) yy, and should not exceed 3 words."

Passage #1 Title: {Passage #1 Title}
Passage #1 Text: {Passage #1 Text}

...

Passage #N Title: {Passage #N Title}
Passage #N Text: {Passage #N Text}

Question: {Question}

Answer:
```

A.2 Prompt Level Rerank Framework Implementation (Zhuang et al., 2024a)

Since Zhuang et al. (2024a) did not release their implementation, we re-implemented their prompt-level rerank framework based on the key ideas outlined in their work. Listing 2 shows the prompt used to rerank passages with LLMs, which includes the query and document text, followed by a relevance judgment instruction on a scale from 0 (Not Relevant) to 4 (Perfectly Relevant), with the output constrained to an integer.

Prompt for Passage Relevance Reranking

Document: {doc_text}

From a scale of 0 to 4, judge the relevance between the query and the document.

0 means 'Not Relevant', 1 means 'Little Relevant', 2 means 'Somewhat Relevant', 3 means 'Highly Relevant', 4 means 'Perfectly Relevant'.

Return only the integer.
```
Query: {query}

Document: {doc_text}

From a scale of 0 to 4, judge the relevance between the query and the document.

0 means 'Not Relevant', 1 means 'Little Relevant', 2 means 'Somewhat Relevant', 3 means 'Highly Relevant', 4 means 'Perfectly Relevant'.

Return only the integer.
```

B Theoretical Discussion on Embedding Space

For a give set of bounded vectors $\{v_i\}$, the orthogonality can be guaranteed by minimizing the following variable (Jain et al., 2014):

$$\epsilon = \mathbf{max}_{i \neq j} |v_i \cdot v_j|^2$$

In our work, we suppose the vectors sampled from a LLM are v, and the vectors introduced by injecting embeddings to the LLM are u. Since there are multiple tokens, including the input tokens and the token will be generated from the LLM, we mark the tokens' vectors as v_i. Then, we have an equivalent definition under LLM setup:

$$\epsilon' = \mathbf{max}_{i \neq j, u} \{ |v_i \cdot v_j|^2, |v_i, u|^2 \}$$

If we have $\forall v_i$, $|v_i, u|^2 \leq \mathbf{max}_{i,j} |v_i \cdot v_j|^2$, obviously we have $\epsilon = \epsilon'$. Otherwise, we obtain a large ϵ by injecting the embedding. Therefore, we only need to minimize the $\mathbf{max}_{i,u}\{|v_i, u|^2\}$.

According to existing theoretical analysis (Geshkovski et al., 2024), we assume that the v_i is a k dimensional Gaussian vector with mean μ_v and variance σ_v on each dimension. (Empirically, $\mu_v \to 0$ and $\sigma_v \to \varepsilon$, where ε is a small number.)

Then, for any given injected vector u, the orthogonality should be decided by $|v_i, u|^2$. However, the v_i contains not only the input tokens' representation but also the prediction of the future. It is computationally expensive to pick a reasonable u after the whole decoding process. It is also meaningless to do so since we have already obtained the decoding results and no need to discuss whether the injected token embedding is efficient or not.

So, the problem is, how to estimate the potential inner product with all possible token embeddings for a given vector u. Here, we propose to use order statistics to find an equivalent regularization.

We first sort the values in each dimension of u into descending order:

$$u' = \{u_{(1)}, u_{(2)}, u_{(3)}, ..., u_{(k)}\}$$

where

$$u_{(1)} \geq u_{(2)} \geq u_{(3)} \geq ... \geq u_{(k-1)} \geq u_{(k)}$$

Then we duplicate the swapping operation on v, which means if we swap the i-th dimension with j-th dimension on u, then the same operation will be deployed on v as well. Here, we suppose there is a general distribution of v on token embeddings and we mark the duplicating operation results as:

$$v' = \{v_{(1)}, v_{(2)}, v_{(3)}, ..., v_{(k)}\}$$

Considering the space between two adjacent variables in order statistics $\Delta_i = u_{(i)} - u_{(i+1)}$ with the only large half where $1 \leq i \leq k/2$, we have:

$$\Sigma_{i=1}^{k/2} v_{(i)} u_{(i)} = \Sigma_{i=1}^{k/2} \Sigma_{j=1}^{i} \Delta_i v_{(j)}$$

According to (Boucheron and Thomas, 2012), we have:

$$\mathbb{E}(\Delta_i^2) \leq \frac{2}{i^2} \mathbb{E}(e^{\frac{u_{(i)}^2}{2}} \int_{u_{(i)}}^{+\infty} e^{-\frac{t^2}{2}} dt)$$

Obviously, there are three parts in the expectation of space Δ_i in the order statistics $u_{(i)}$. With the increasing of order index i in $u_{(i)}$, the value of $u_{(i)}$ is decreasing. Thus, the $2/i^2$ will converge to zero and the experiential term $e^{u_i^2/2}$ will decrease to 1 since most embedding in LLM are around the original point in current popular models like LLama or Qwen. For the third term of $\int_{u_{(i)}}^{+\infty} e^{-\frac{t^2}{2}} dt$, it is a survival function from Gaussian distribution, then it should be bounded by 1. (In fact, it should be bounded by 0.5 if we only consider the positive $u_{(i)}$ from the first half of order statistics)

Therefore, for $\forall s > 0$, $\exists \varepsilon > 0$, when $i > s$, we have $\mathbb{E}(\Delta_i^2) < \varepsilon$ is true.

Meanwhile, for the whole inner product term, we have

$$|\Sigma_{i=1}^{k} v_{(i)} u_{(i)}|^2 < |\Sigma_{i=1}^{\frac{k}{2}} v_{(i)} u_{(i)}|^2 + |\Sigma_{i=\frac{k}{2}+1}^{k} v_{(i)} u_{(i)}|^2$$

where we consider both sides of the order statistics which may contain a large absolute value but negative u_i.

Then, considering the bound of $\mathbb{E}(\Delta_i^2)$, and $v_{(j)}$ follows a Gaussian distribution with 0 mean for all dimensions, then the expectation of inner product highly relies on the first top s element in the ordered statistics, where:

$$|\Sigma_{i=1}^{k} v_{(i)} u_{(i)}|^2 \leq 4\sigma_v^2 \Sigma_{i=1}^{s} \Delta_i^2 \mathbb{E}(e^{\frac{u_{(i)}^2}{2}}) + 2s\sigma_v^2 \varepsilon$$

Therefore, in our proposed method, we design a spacing-based method according to the top s order statistics to approximate the orthogonality of the whole embedding space.

C Additional Results

C.1 EmbQA Performance Comparison between Prompt Level Rerank framework and Embedding Level Rerank Framework

Our experiments (Table 6) demonstrate that replacing the prompt-level rerank framework with our embedding-level alternative consistently yields superior Exact Match and F1 scores across BM25, DPR, and Contriever on HotpotQA, 2Wiki, NQ, and WebQ, thereby confirming the effectiveness and efficiency of our embedding-level rerank approach.

Retriever& Rerank Framework	HotpotQA		2Wiki		NQ		WebQ	
	EM	F1	EM	F1	EM	F1	EM	F1
BM25	25.4	37.15	16.6	21.11	26.0	32.81	22.2	31.23
+Prompt Level	39.6	52.35	19.8	28.67	39.8	51.83	36.6	50.05
+Embedding Level (Ours)	**42.0**	**55.81**	**27.4**	**36.60**	**42.2**	**54.38**	**38.2**	**52.08**
DPR	20.6	21.67	10.8	13.53	25.0	34.16	23.8	34.44
+Prompt Level	24.6	29.99	11.6	18.84	39.8	52.28	37.6	48.23
+Embedding Level (Ours)	**29.8**	**36.34**	**16.8**	**21.03**	**43.0**	**54.36**	**38.0**	**51.95**
Contriever	22.6	35.42	16.6	20.67	25.8	32.83	25.2	34.17
+Prompt Level	30.2	47.26	17.6	25.92	39.8	52.21	34.6	48.67
+Embedding Level (Ours)	**36.6**	**52.68**	**26.4**	**34.22**	**42.2**	**53.58**	**36.0**	**49.60**

Table 6: EmbQA Overall Performance on HotpotQA, 2Wiki, NQ, and WebQ datasets using LLaMA3.1.