Improving Long-Context Retrieval with Multi-Prefix Embedding

Anonymous ACL submission

Abstract

Long-context retrieval exposes a tension: single-vector embeddings lose fine-grained detail, while token-level multi-vector methods incur prohibitive storage. We propose Multi-Prefix Embedding (MPE), which partitions a document into chunks separated by EOS tokens, encodes the full sequence in a single causal forward pass, and extracts one embedding at each prefix boundary. MPE retains cross-chunk context, enables chunk-level MaxSim matching, and trains with only document-level relevance labels. Experiments on MLDR-en, BrowseComp-Plus, and LongEmbed show that MPE is competitive with or outperforms single-vector, independent-chunk, and multi-vector baselines, while providing a natural source attribution mechanism for locating evidence chunks.

1 Introduction

Long-context retrieval requires models to identify localized evidence within documents that may span thousands of tokens. Dense retrieval encodes queries and documents as fixed-dimensional vectors for efficient inner-product search (Karpukhin et al., 2020; Izacard et al., 2022; Qwen Team, 2025; Xu et al., 2025), but compressing a long document into a single vector can obscure fine-grained evidence (Zhu et al., 2024). Token-level multi-vector methods such as ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022) preserve finer granularity, but incur substantially larger indexes and higher retrieval cost. Chunk-based retrieval is cheaper, but independently encoded chunks lose cross-chunk context and require post-hoc document-level aggregation. This raises a central question: can we retain fine-grained matching while preserving context and keeping the representation compact?

We propose *Multi-Prefix Embedding* (MPE), a compact multi-vector representation for causal embedding models. As shown in Figure 1, MPE splits a long document into chunks, inserts EOS tokens between them, and encodes the resulting sequence in a single causal forward pass. The hidden state at each EOS position summarizes the preceding content, yielding one prefix embedding per chunk boundary without architectural changes or chunk-level annotations. Query–document similarity is computed with MaxSim over the document's prefix embeddings, and the model is trained using only document-level relevance labels.

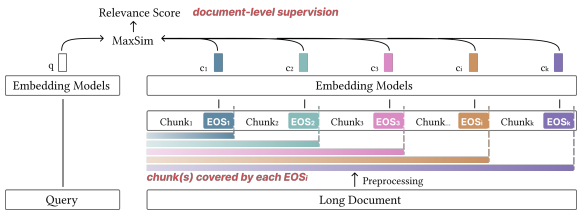

Figure 1: Multi-Prefix Embedding (MPE). A long document is split into chunks separated by EOS tokens and encoded in one causal forward pass. Prefix embeddings are extracted at $\text{EOS}_1, \ldots, \text{EOS}_K$, and query–document relevance is computed via MaxSim.

MPE occupies a middle ground between single-vector, independent-chunk, and token-level multi-vector retrieval. It provides multiple matching points within a long document, preserves preceding cross-chunk context through causal attention, and stores only one embedding per chunk boundary rather than one embedding per token. The MaxSim-selected prefix also identifies the document region most responsible for a match, providing a lightweight mechanism for surfacing candidate evidence.

We evaluate MPE on MLDR-en (Chen et al., 2024), BrowseComp-Plus (Chen et al., 2025), and LongEmbed (Zhu et al., 2024). Fine-tuning Qwen3-Embedding-0.6B (Qwen Team, 2025) on MLDR-en, MPE matches or exceeds single-vector and independent-chunk baselines on most benchmarks, while random prefix-length augmentation improves robustness under granularity mismatch, especially on out-of-domain BrowseComp-Plus.

Our main contributions are:

1. We propose MPE, a compact multi-vector representation for long-context retrieval that balances

fine-grained matching, context preservation, and indexing efficiency.

2. We show that MaxSim-selected prefixes often align with annotated answer locations, suggesting a lightweight mechanism for candidate evidence surfacing.
3. We present experiments and ablations across representation methods, attention mechanisms, and granularity settings.

2 Related Work

Dense retrieval encodes queries and documents into fixed-dimensional vectors for inner-product search (Karpukhin et al., 2020; Izacard et al., 2022). DPR uses BERT encoders with [CLS] pooling (Karpukhin et al., 2020), while recent decoder-only embedding models such as RepLLaMA (Ma et al., 2024), E5-Mistral (Wang et al., 2024), and Qwen3-Embedding (Qwen Team, 2025) use last-token pooling. Although efficient, single-vector embeddings can obscure fine-grained evidence in long documents.

Multi-vector methods improve fine-grained matching by representing each document with multiple embeddings. ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022) uses late interaction over token-level representations, but its storage and retrieval cost grows with document length. MaxP (Dai and Callan, 2019) splits documents into fixed-size chunks and takes the maximum independently encoded chunk score, but loses cross-chunk context.

MPE is most related to compact and contextual multi-vector representations. BGE Landmark Embedding (Luo et al., 2024) also extracts hidden states from special positions in a causal LLM, but targets context selection for retrieval-augmented long-context LLMs using landmark tokens and multi-stage supervision. In contrast, MPE uses EOS-separated chunk boundaries for document-level retrieval and trains with a single contrastive objective over document-level labels. Late chunking (Günther et al., 2024) pools chunk representations after bidirectional document encoding, while token pooling (Clavié et al., 2024) compresses ColBERT-style token embeddings for short-passage retrieval.

3 Method

MPE consists of EOS-based prefix construction, MaxSim scoring, and random prefix-length augmentation. Figure 1 illustrates the overall framework.

Given a document $p = [t_1, \ldots, t_L]$, we split it into K consecutive chunks of m content tokens and append an EOS token after each chunk:

$$x = [\, t_1 \ldots t_m, \text{<eos>}, \; t_{m+1} \ldots t_{2m}, \text{<eos>} \ldots] \quad (1)$$

The resulting sequence is encoded in a single forward pass through a pretrained causal language model. Under causal masking, the hidden state at each EOS position attends to the current chunk and all preceding chunks, yielding a contextualized prefix representation. Unlike independently encoded chunks, each prefix embedding therefore incorporates preceding cross-chunk context. We L2-normalize the hidden states at EOS positions to obtain document prefix embeddings $\{c_1, \ldots, c_K\}$, where $c_k \in \mathbb{R}^d$. When $K = 1$, MPE reduces to standard last-token pooling.

3.1 Scoring, Training, and Retrieval

Given a query embedding q, MPE scores a document by MaxSim over its prefix embeddings:

$$s(q, p) = \text{MaxSim}(q, p) = \max_{k \in \{1, \ldots, K\}} q^\top c_k \quad (2)$$

This allows a query to match the most relevant document region rather than requiring the whole document to be compressed into one vector. We train MPE with contrastive loss over document-level relevance labels using cross-device in-batch negatives.

At retrieval time, all prefix embeddings are indexed in a FAISS (Johnson et al., 2019) flat inner-product index together with their document IDs. Prefix-level hits are aggregated by document ID using the maximum prefix score to produce the final document ranking.

3.2 Random Prefix-Length Augmentation

Fixed chunk sizes can make the model sensitive to a specific granularity. To improve robustness, we sample the content chunk length for each training passage, $m \sim \mathcal{U}[m_{\min}, m_{\max}]$, and construct EOS-separated prefixes using that length. All chunks within a passage share the same sampled m. This exposes the model to diverse prefix boundaries during training and helps a single MPE model generalize across inference chunk sizes without retraining.

4 Experiments

4.1 Setup

Datasets. We evaluate on three long-context retrieval benchmarks. *MLDR-en* (Chen et al., 2024) contains long-document retrieval queries with documents up to 8,192 tokens; we use its 800-query test split. *BrowseComp-Plus* (Chen et al., 2025) contains 830 difficult web queries with human-verified supporting documents and mined hard negatives. *LongEmbed* (Zhu et al., 2024) evaluates long-context retrieval ability; we report results on the real-document subsets NarrativeQA, 2WikiMQA, SummScreen, and QMSum.

Model and training. We use Qwen3-Embedding-0.6B (Qwen Team, 2025) as the base model and fine-tune it on the MLDR-en training split for one epoch using Tevatron (Ma et al., 2025). Training uses LoRA adapters with rank 16, $\alpha = 64$, and dropout 0.1 on all attention and MLP projections (`q/k/v/o/gate/up/down_proj`). We use learning rate 1×10^{-4}, temperature $\tau = 0.03$, per-GPU batch size 2, group size 4, query length 512, maximum document length 8,192, right-side padding, and gradient checkpointing. Experiments were run on 8 NVIDIA RTX 5090 GPUs. The total compute budget for the reported experiments is approximately 120 GPU-hours, including training, ablations, and inference.

For zero-shot evaluation, we reuse the MLDR-en fine-tuned model without further training. We use maximum document length 4,096 for BrowseComp-Plus and 8,192 for LongEmbed.

Configurations. We compare five configurations that isolate the effects of multi-vector inference, MaxSim training, cross-chunk attention, and random prefix-length augmentation (Table 1).

Setting	MaxSim Train	Cross-Chunk Attn.	Rand. Size
Single-vector	×	–	×
MaxP	×	×	×
MaxP-Train	✓	×	×
MPE Fixed-N	✓	✓	×
MPE-Rand$[a, b]$	✓	✓	✓

Table 1: Comparison of configuration settings.

Single-vector uses last-token pooling without chunking. *MaxP* (Dai and Callan, 2019) encodes chunks independently at inference and takes the maximum chunk score. *MaxP-Train* adds MaxSim training. *MPE Fixed-N* trains and infers with fixed EOS-separated prefixes, while *MPE-Rand*$[a, b]$ samples the training chunk size from $[a, b]$. Unless specified, chunk-based and MPE methods use inference chunk size 64; LongEmbed and BrowseComp-Plus are zero-shot after MLDR-en training.

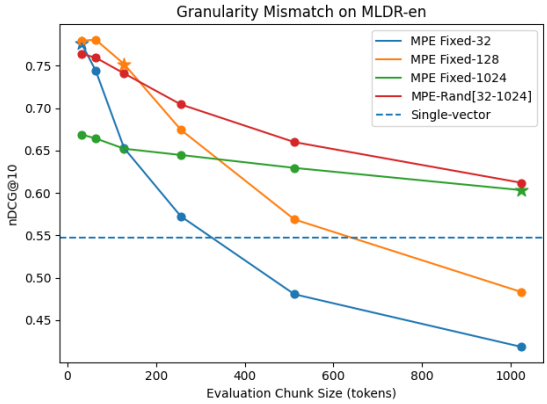

Figure 2: Granularity mismatch on MLDR-en. Fixed-size MPE degrades under mismatched granularities, while MPE-Rand tracks the upper envelope with a single model. *Star* symbols denote matched train–eval sizes; *circle* symbols denote mismatched sizes.

4.2 Main Results

Table 2 reports nDCG@10 on in-domain and zero-shot long-context retrieval benchmarks. MPE Fixed-64 achieves the best dense result on MLDR-en, while MPE-Rand$[32, 1024]$ is the only multi-vector variant to outperform Single-vector on BrowseComp-Plus, highlighting the importance of granularity robustness for out-of-domain retrieval. On LongEmbed, MPE gives the best dense result on 2WikiMQA and QMSum and remains competitive on NarrativeQA; SummScreen is nearly saturated by BM25 and Single-vector retrieval.

MPE improves over independent chunking through contextual prefix embeddings and granularity robustness. Under the same inference chunk size, MPE Fixed-64 outperforms MaxP-Train on MLDR-en, suggesting that preceding cross-chunk context helps. MPE-Rand sacrifices some in-domain peak performance on MLDR-en but improves BrowseComp-Plus and QMSum.

Granularity robustness. Figure 2 varies the inference chunk size from 32 to 1024 on MLDR-en. Fixed-size MPE models peak near their training granularity but degrade under mismatch, while MPE-Rand$[32, 1024]$ tracks their upper envelope with a single checkpoint.

Causal vs. bidirectional attention. As a diagnostic, we replace the causal self-attention mask

Method	MLDR Train	MLDR-en	BrowseComp-Plus	NarrativeQA	2WikiMQA	SummScreen	QMSum
BM25	×	0.679	0.016	0.715	0.965	0.976	0.813
jina-v2-base-en (Günther et al., 2023)	×	0.370	–	0.394	0.740	0.935	0.389
OpenAI-Ada-002	–	0.387	–	0.411	0.801	0.918	0.400
E5-Mistral-7B (Wang et al., 2024)	×	0.433	–	0.449	–	–	–
BGE-M3 (Chen et al., 2024)	✓	0.489	–	0.487	0.780	0.940	0.355
BGE-M3$_{\text{mv}}$ (Chen et al., 2024)	✓	0.558	–	0.554	–	–	–
Single-vector	✓	0.548	0.132	0.543	0.863	**0.981**	0.565
MaxP	✓	0.758	0.103	**0.604**	0.944	0.906	0.550
MaxP-Train	✓	0.776	0.104	0.557	0.945	0.903	0.570
MPE Fixed-64	✓	**0.783**	0.122	0.580	**0.948**	0.928	0.652
MPE-Rand$[32, 1024]$	✓	0.760	**0.153**	0.579	0.945	0.949	**0.705**

Table 2: nDCG@10 on long-context retrieval benchmarks. Baseline results are from original papers (Günther et al., 2023; Zhu et al., 2024; Chen et al., 2024; Wang et al., 2024; Nussbaum et al., 2024). BGE-M3$_{\text{mv}}$ denotes ColBERT-style multi-vector retrieval. Methods below the midrule use Qwen3-Embedding-0.6B; chunk-based and MPE methods use inference chunk size 64 unless otherwise specified. Bold marks the best dense result.

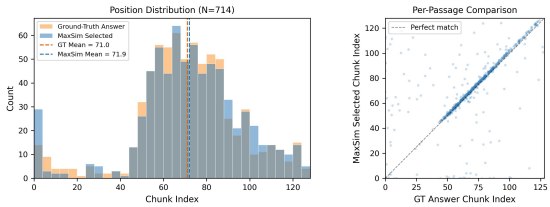

Figure 3: MaxSim-selected chunk positions vs. Gemini-annotated answer positions on MLDR-en. Left: overlaid position distributions. Right: per-passage scatter plot showing substantial rank correlation (Spearman $\rho = 0.77$).

Retriever	Acc. (%)	Recall (%)	Avg. Calls	Cal. Err. (%)[1]
BM25	34.94	37.78	18.85	51.17
Single-vector	42.29	48.00	17.40	46.99
MPE-Rand	**51.45**	**57.70**	**15.78**	**39.74**

Table 3: End-to-end BrowseComp-Plus with a Gemini 3 Flash Preview search agent. Accuracy is judged by GPT-4o-mini. MPE-Rand is trained on $[32, 1024]$ and evaluated at chunk size 64.

with a bidirectional mask during fine-tuning and inference. This modification reduces MLDR-en nDCG@10 from 0.783 to 0.694, suggesting that MPE benefits from the pretrained causal attention structure rather than simply from exposing each EOS position to more tokens.

Storage overhead. MPE stores $K = T/C$ vectors per document, where T is document length and C is chunk size. On MLDR-en ($T \leq 8{,}192$), this gives $K \approx 128$ at $C = 64$ and $K \approx 16$ at $C = 512$. For N documents, MPE stores $\mathcal{O}(KN)$ vectors, between single-vector retrieval $\mathcal{O}(N)$ and token-level late interaction $\mathcal{O}(TN)$, with $K \ll T$.

Source attribution. Using Gemini-annotated answer spans on 714 MLDR-en passages, MaxSim selects a chunk within ± 1 position of the annotated chunk for 65.7% of passages, with Spearman correlation $\rho = 0.77$ (Figure 3). Although the annotations are LLM-derived, this suggests that MPE can surface candidate evidence in addition to document-level rankings.

End-to-end search agent. Table 3 evaluates MPE-Rand in a BrowseComp-Plus search-agent setting. Replacing the single-vector retriever with MPE-Rand improves answer accuracy from 42.29% to 51.45%, supporting-document recall from 48.00% to 57.70%, and reduces retrieval calls.

5 Conclusion

We presented Multi-Prefix Embedding (MPE), a compact chunk-level multi-vector method for long-document retrieval. By inserting EOS tokens between chunks, MPE extracts prefix embeddings in a single causal LM forward pass, preserving preceding cross-chunk context while enabling fine-grained MaxSim matching without chunk-level supervision or architectural changes.

Experiments on MLDR-en, BrowseComp-Plus, and LongEmbed show that MPE matches or outperforms single-vector and independent-chunk baselines on most benchmarks. Random prefix-length augmentation improves robustness across chunk sizes, with particularly strong gains under granularity mismatch and out-of-domain retrieval. MaxSim-selected prefixes align with Gemini annotated answer locations.

Code and model checkpoints will be released upon publication.

[1] Calibration error measures how closely predicted confidence matches actual answer accuracy (Chen et al., 2025).

Limitations

Our experiments are limited to a single 0.6B-parameter causal embedding model and English-language benchmarks. Scaling MPE to larger embedding models, multilingual retrieval, and stronger instruction-tuned retrievers remains future work.

MPE introduces additional storage and retrieval cost compared with single-vector embeddings because each document is represented by multiple prefix embeddings. While this cost is substantially smaller than token-level late interaction, practical deployment may require product quantization, residual compression, pruning, or adaptive prefix selection.

Our comparisons against Landmark Embedding (Luo et al., 2024) and Late Chunking (Günther et al., 2024) are based on reported results and conceptual differences rather than a fully controlled head-to-head evaluation. A matched comparison under the same backbone, training data, context length, and indexing budget would better isolate the effect of MPE's prefix representation.

Finally, our source-attribution analysis relies on Gemini-generated answer-span annotations. Although the observed correlation suggests that MaxSim often selects meaningful evidence locations, these labels are not human gold annotations and may contain noise.

Ethics Descriptions

This work uses public retrieval benchmarks and does not involve human-subject data collection. Like other retrieval systems, MPE may amplify biases or misinformation present in training corpora or retrieved documents, especially when used in downstream search or question-answering agents. Our BrowseComp-Plus agent evaluation uses LLM-based answer judging, which can introduce noise or evaluator bias; we therefore treat it as complementary to retrieval metrics rather than definitive human evaluation. We disclose the main compute setup in Section 4.1; practical deployments should consider the storage, latency, and energy costs of multi-vector indexing.

References

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*.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. BrowseComp-Plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Benjamin Clavié, Antoine Chaffin, and Griffin Adams. 2024. Reducing the footprint of multi-vector retrieval with minimal performance impact via token pooling. *arXiv:2409.14683*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 985–988, New York, NY, USA. Association for Computing Machinery.

Michael Günther, Isabelle Mohr, Daniel J Williams, Bo Wang, and Han Xiao Rodríguez. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Afzal Akber, Susana Gusber, Georgios Mastrapas, Robin Milliken, and Bo Wang. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Kun Luo, Zheng Liu, Shitao Xiao, and Kang Liu. 2024. BGE landmark embedding: A chunking-free embedding method for retrieval augmented

long-context large language models. *arXiv preprint arXiv:2402.11573*.

Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. 2025. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. *arXiv preprint arXiv:2505.02466*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2421–2425, New York, NY, USA. Association for Computing Machinery.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

Qwen Team. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Zhichao Xu, Fengran Mo, Zhiqi Huang, Crystina Zhang, Puxuan Yu, Bei Wang, Jimmy Lin, and Vivek Srikumar. 2025. A survey of model architectures in information retrieval. *arXiv preprint arXiv:2502.14822*.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. Longembed: Extending embedding models for long context retrieval. *arXiv preprint arXiv:2404.12096*.

A License

We use the following external artifacts: MLDR-en, BrowseComp-Plus, LongEmbed, and Qwen3-Embedding-0.6B. MLDR-en and BrowseComp-Plus are released under the MIT license. Qwen3-Embedding-0.6B is released under the Apache-2.0 license. LongEmbed is a public research benchmark for long-context retrieval evaluation; its Hugging Face dataset card does not list an explicit license field, so we use it only for research evaluation and follow the terms and licenses of the original component datasets included in LongEmbed.

Our use of these artifacts is consistent with their intended research purpose: datasets are used for retrieval benchmarking and evaluation, and the pretrained Qwen3-Embedding model is used as a base embedding model for research experiments. We do not redistribute the original benchmark data beyond what is permitted by the artifact providers. Any released code or checkpoints from this work are intended for research and reproducibility, and users should comply with the licenses and terms of the underlying datasets, pretrained models, and software dependencies.

B LLM Usage

The authors used generative AI tools for language polishing, grammar checking, minor editing suggestions, and assistance with code debugging. The tools were not used to generate the core technical ideas, design the experiments, produce the reported results, or make final scientific claims. All experiments, analyses, results, and final writing decisions were performed and verified by the authors.