Embedding-based In-Context Prompt Training for Enhancing LLMs as Text Encoders

Ailiang Lin[1], **Zhuoyun Li**[2], **Keyu Mao**[1], **Kotaro Funakoshi**[1], **Manabu Okumura**[1]
[1]Institute of Science Tokyo [2]Tencent
{linailiang, maokeyu, funakoshi, oku}@lr.first.iir.isct.ac.jp
earyli@tencent.com

Abstract

Large language models (LLMs) have been widely explored for embedding generation. While recent studies show that in-context learning (ICL) effectively enhances the representational capability of LLMs by prepending a few task-related demonstrations, it causes substantial token overhead due to the increased sequence length. In this work, we propose EPIC, a novel embedding-based in-context prompt training strategy that leverages ICL to generate high-quality embeddings while reducing computational burden during both training and inference. This approach replaces discrete text demonstrations with their corresponding continuous embeddings, which not only encourages the LLM to align semantically-related text pairs during contrastive learning, but also requires the model to interpret demonstration embeddings as part of the in-context prompt. Consequently, EPIC-trained models achieve excellent embedding performance both with or without in-context prompts at inference time. Comprehensive experiments demonstrate that our method establishes new state-of-the-art results on the MTEB benchmark, surpassing frontier models trained solely on publicly available retrieval data. Extensive ablation studies further validate the effectiveness and necessity of our mechanism.

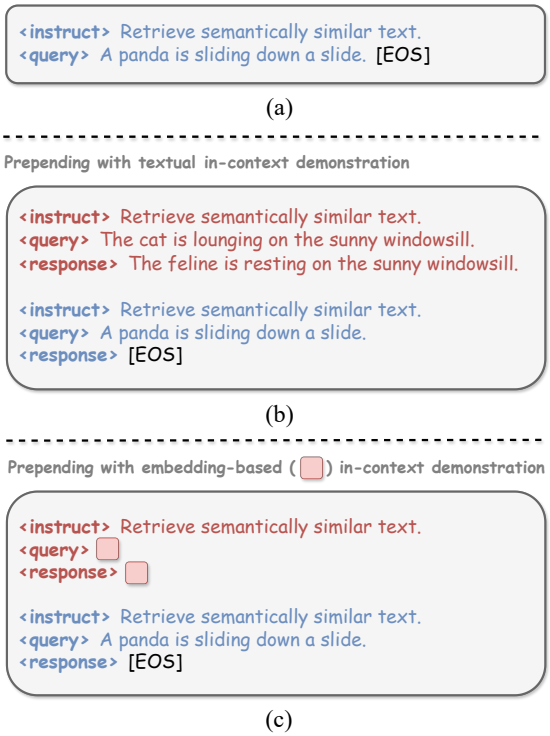


Figure 1: Comparison of different inputs for embedding tasks. (a) Embedding models typically take only the task instruction and user query as input. (b) Li et al. (2025) adopt the in-context learning strategy by incorporating task-related demonstrations. (c) EPIC enhances the input by prepending it with an embedding-based in-context prompt.

1 Introduction

Text embeddings are powerful vector representations that capture contextual semantics of variable-length texts, playing a critical role in various natural language processing (NLP) tasks (Muennighoff et al., 2023). For example, retrieval-augmented generation (RAG) systems typically encode textual queries and documents into a shared embedding space, enabling efficient retrieval through similarity search (Lewis et al., 2020; Liu et al., 2024b).

The rapid progress of Large Language Models (LLMs) brings new possibilities for improving the quality of text embeddings. Given the remarkable semantic understanding capabilities showcased by LLMs, recent research (Muennighoff et al., 2024; BehnamGhader et al., 2024; Springer et al., 2025; Lee et al., 2025a; Pan et al., 2025; Su et al., 2025) has increasingly focused on adapting them into text encoders through supervised contrastive learning (Gao et al., 2021; Wang et al., 2024a).

In particular, PromptEOL (Jiang et al., 2024) incorporates in-context learning (ICL) (Brown et al., 2020) into text embedding in a training-free manner. However, Muennighoff et al. (2024) show that

Findings of the Association for Computational Linguistics: ACL 2026, pages 29079–29095

ICL cannot be directly applied to fine-tuned embedding models. To overcome this limitation, bge-en-icl (Li et al., 2025) introduces a simple training strategy that effectively endows embedding models with ICL capabilities by prepending a few task-related *query-passage* pairs (a.k.a. *query–response* pairs) as demonstrations to the input text during contrastive learning. While these approaches highlight the potential of leveraging ICL to enhance text representation learning, their in-context demonstrations remain restricted to the discrete textual form, which substantially increases the input length and imposes a heavy token burden during training and inference, making them less practical in latency-sensitive scenarios, such as information retrieval and RAG tasks. Meanwhile, recent studies (Hendel et al., 2023; Zhuang et al., 2024) suggest that the ICL capabilities of LLMs can be extended to continuous vector representations under the next-token prediction paradigm, opening new avenues for more efficient exploitation of ICL.

In this context, we propose an **E**mbedding-based **P**rompt training with **I**n-**C**ontext demonstrations (**EPIC**), which leverages ICL to enhance the representational capability of LLMs while reducing computational overhead during both training and inference. Specifically, as shown in Figure 1, we replace textual in-context demonstrations with their vector representations to form the embedding-based in-context prompt, which is then concatenated with the input query to obtain the desired query embedding. Since both the in-context and query embeddings are generated by the same model, contrastive learning not only encourages the LLM to align semantically-related positive pairs but also requires it to interpret the demonstration embeddings as part of the in-context prompt. During training, the demonstrations are directly sampled from the embeddings of positive pairs within the same batch. At inference time, we can pre-compute and reuse the embedding-based in-context prompts, avoiding redundant attention computation on textual demonstrations and thereby reducing inference latency.

We evaluate our EPIC on the Massive Text Embeddings Benchmark (MTEB) (Muennighoff et al., 2023) across three popular LLMs, including Qwen2.5-7B, Mistral-7B, and LLaMA-3.1-8B. Experimental results show that our method achieves embedding performance on par with models trained with discrete textual ICL. Moreover, we observe an intriguing representational property: even without any in-context prompts during inference, the EPIC-trained models outperform the conventionally trained baselines under the same conditions. Notably, the proposed EPIC achieves new state-of-the-art results on MTEB among models trained exclusively on publicly available retrieval data. Extensive ablation studies further confirm the effectiveness and necessity of our approach.

The primary contributions of this work are summarized as follows:

- We propose EPIC, a novel embedding-based in-context prompt training strategy that enhances LLMs as text encoders while reducing token overhead compared to textual ICL.
- Experimental results demonstrate that LLMs trained with EPIC consistently improve embedding performance even without in-context demonstrations during inference.
- EPIC-trained models achieve new state-of-the-art results on MTEB. We further provide in-depth ablation studies to validate the effectiveness and necessity of our method.

2 Method

In this section, we first introduce the preliminaries of conventional in-context learning (ICL) for text embedding in Section 2.1. We then present our embedding-based in-context prompt (EPIC) method in Section 2.2. Finally, we describe the training and inference strategies based on EPIC in Sections 2.3 and 2.4, respectively.

2.1 Preliminary

For LLM-based embedding models, the text embedding is typically derived from the final hidden state of the special end-of-sequence (EOS) token, since only the last token can access the full sequence context under the causal attention mechanism. Specifically, given an input sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$ of length n with embedding dimension d, in addition to appending the `[EOS]` token, we prepend a task-specific instruction $\mathbf{I}$, which enables the model to generalize across different embedding tasks (Wang et al., 2024a). The vector representation of the input text is formally defined as:

$$\mathbf{e}_{\mathbf{X}} = f_\theta^{\text{EOS}}([\mathbf{I}; \mathbf{X}; [\texttt{EOS}]]) \in \mathbb{R}^d, \qquad (1)$$

where $[\cdot;\cdot]$ denotes the sequence concatenation operation and $f_\theta^{\text{EOS}}(\cdot)$ refers to a function that returns the final hidden state of the LLM for the last input token, i.e., `[EOS]`.

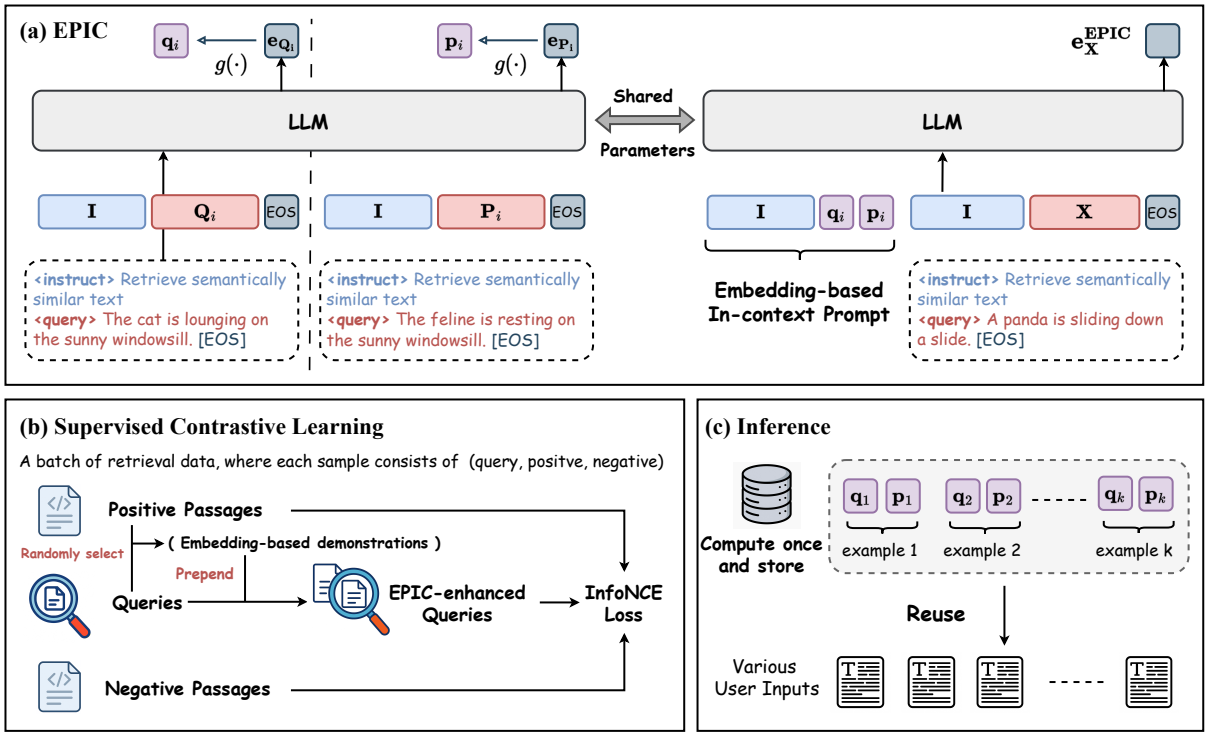


Figure 2: (a) Overview of the proposed EPIC method. For a given task (e.g., STS), the user input is "*A panda is sliding down a slide*", while the demonstration query–passage pair consists of "*The cat is lounging on the sunny windowsill*" and "*The feline is resting on the sunny windowsill*". (b) During training, we randomly sample (`query`, `positive`) embedding pairs from the same batch as in-context demonstrations, which are then used to construct EPIC-enhanced `queries`. (c) The demonstration embeddings are pre-computed once and reused at inference time.

Considering that the instruction alone provides limited information, bge-en-icl (Li et al., 2025) expands the input sequence with a k-shot demonstration set $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_k\}$ to integrate the in-context learning (ICL) capabilities (Brown et al., 2020) of LLMs into text embeddings. Concretely, each demonstration $\mathbf{D}_i$ consists of an instruction and a task-related *query–passage* pair, i.e., $\mathbf{D}_i = [\mathbf{I}; \mathbf{Q}_i; \mathbf{P}_i]$, as illustrated in Figure 1(b). The ICL-based text embedding can be computed as:

$$\mathbf{e}_{\mathbf{X}}^{\text{ICL}} = f_\theta^{\text{EOS}}([\mathbf{D}_1; \mathbf{D}_2; \ldots; \mathbf{D}_k; \mathbf{I}; \mathbf{X}; [\texttt{EOS}]]). \quad (2)$$

Notably, directly adding few-shot demonstrations in the prompts is generally ineffective for standard fine-tuned embedding models (Muennighoff et al., 2024). Therefore, ***the ICL in bge-en-icl and throughout the following discussion refers to capabilities acquired through specialized training strategies***, rather than the original formulation without any gradient updates.

2.2 Embedding-based In-Context Prompt

While ICL has been shown to significantly enhance embedding quality (Jiang et al., 2024; Li et al., 2025), conventional in-context demonstrations introduce a large number of extra text tokens, leading to substantial computational overhead. This raises an intriguing question: *could the embedding model benefit from ICL while mitigating the surge in sequence length*?

Inspired by the proven effectiveness of text embeddings, which inherently encode the contextual semantics of text, we challenge conventional wisdom by proposing an **E**mbedding-based **P**rompt training strategy with **I**n-**C**ontext demonstrations (**EPIC**) to improve the representational capacity of LLMs as text encoders. Specifically, as shown in Figure 2(a), rather than using discrete textual demonstrations, we replace each *query–passage* pair $(\mathbf{Q}_i, \mathbf{P}_i)$ with its corresponding continuous text embeddings. To further align these embedding-based demonstrations, we introduce a lightweight MLP layer $g(\cdot)$ consisting of two linear transformations with a GELU activation. The resulting continuous vector representations of the in-context *query-passage* pair are computed as:

$$\begin{aligned} \mathbf{q}_i &= g(f_\theta^{\text{EOS}}([\mathbf{I}; \mathbf{Q}_i; [\texttt{EOS}]])) \in \mathbb{R}^d, \\ \mathbf{p}_i &= g(f_\theta^{\text{EOS}}([\mathbf{I}; \mathbf{P}_i; [\texttt{EOS}]])) \in \mathbb{R}^d. \end{aligned} \quad (3)$$

The two vectors $\mathbf{q}_i$ and $\mathbf{p}_i$ compress the discrete *query–passage* pair $(\mathbf{Q}_i, \mathbf{P}_i)$ into a shared latent space, substantially reducing token usage, since $|\mathbf{Q}_i| + |\mathbf{P}_i| \gg 2$, where $|\cdot|$ denotes the sequence length. Accordingly, we transform the textual demonstration set $\mathcal{D}$ into an embedding-based version $\mathcal{E} = \{\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_k\}$, where each $\mathbf{E}_i = [\mathbf{I}; \mathbf{q}_i; \mathbf{p}_i]$. Consequently, the EPIC-enhanced embedding can be expressed as:

$$\mathbf{e}_{\mathbf{X}}^{\text{EPIC}} = f_\theta^{\text{EOS}}([\mathbf{E}_1; \mathbf{E}_2; \ldots; \mathbf{E}_k; \mathbf{I}; \mathbf{X}; [\texttt{EOS}]]). \tag{4}$$

Since the vector representations $\mathbf{q}_i$, $\mathbf{p}_i$, and $\mathbf{e}_{\mathbf{X}}^{\text{EPIC}}$ all originate from the same LLM, which requires the model not only to generate high-quality embeddings but also to interpret its own embeddings when they are fed back as part of the in-context prompt. In this way, EPIC effectively reduces the token overhead of conventional ICL while preserving its representational advantages.

2.3 Supervised Contrastive Learning

In line with previous work (BehnamGhader et al., 2024; Springer et al., 2025), we fine-tune the LLM on publicly available retrieval datasets through contrastive learning, where each training sample consists of a triplet `(query, positive, negative)`. Consequently, each training step involves three forward passes to obtain the corresponding embeddings. To incorporate the proposed EPIC strategy, we perform an additional forward pass to generate the EPIC-enhanced `query` embedding (Figure 2(b)). Following bge-en-icl (Li et al., 2025), we sample different `(query, positive)` embedding pairs from the same batch to construct the embedding-based in-context prompts, which are then used to enhance the original `Query`. The number of demonstration pairs is randomly chosen between 0 and a predefined maximum value, jointly enhancing the model's representational capabilities with and without in-context prompts.

During training, we adopt the standard InfoNCE loss (Izacard et al., 2021), defined as follows:

$$\mathcal{L} = -\log \frac{\phi(q, p^+)}{\phi(q, p^+) + \sum_{d^- \in \mathcal{N}} \phi(q, p^-)}, \tag{5}$$

where (q, p^+) denotes the positive pair and $\mathcal{N}$ represents the set of in-batch and hard negative samples. The function $\phi(\cdot)$ is a temperature-scaled cosine similarity that measures the matching score between two text embeddings, computed as:

$$\phi(q, p) = \exp(\frac{1}{\tau} \cos(\mathbf{e}_q, \mathbf{e}_p)), \tag{6}$$

where τ is a temperature hyperparameter fixed to 0.05 in our experiments.

2.4 Inference

During inference, the proposed EPIC strategy may seem to increase computational cost since it requires generating additional vector representations. However, demonstration embeddings need to be computed only once, and the resulting embedding-based in-context prompt can be reused for the same task (Figure 2(c)). This avoids repeatedly appending lengthy textual demonstrations at inference time, thereby reducing token usage while improving embedding quality.

Furthermore, embedding performance under non-ICL settings is also crucial in practice. As discussed in Section 3.2, we observe a surprising representational effect: even without any in-context prompts during inference, the EPIC-trained models outperform the standard contrastive baselines under the same conditions. In contrast, models trained with conventional ICL do not exhibit such advantages when in-context demonstrations are removed, confirming the practicality of our EPIC.

3 Experiments

3.1 Experimental Setup

Training Datasets. Following BehnamGhader et al. (2024); Li et al. (2025); Pan et al. (2025); Su et al. (2025), we conduct training on the public portion of the E5 dataset (Wang et al., 2024a) curated by Springer et al. (2025). The corpus is a collection of publicly available retrieval datasets, consisting of approximately 1.5M samples. Please refer to Appendix A.2 for more details about the dataset composition.

Training Details. We apply the proposed EPIC to three popular LLMs: Qwen2.5-7B-Instruct (Qwen2.5-7B), Mistral-7B-Instruct-v0.2 (Mistral-7B), and Meta-Llama-3.1-8B-Instruct (LLaMA-3.1-8B). Following the training recipe from bge-en-icl (Li et al., 2025), we fine-tune the models using LoRA (Hu et al., 2022) with rank 64, alpha 32, and a learning rate of $1e^{-4}$. For in-context demonstrations, we randomly sample 0 to 5 (`query, positive`) pairs from the in-batch training data. The maximum sequence length for training is set to 512 tokens. More training details are presented in Appendix A.1.

Categories → # of datasets →	Retr. 15	Rerank. 4	Clust. 11	PairClass. 3	Class. 12	STS 10	Summ. 1	Avg 56
Miscellaneous								
$\text{SimCSE}_{\text{BERT}}$ (Gao et al., 2021)	21.82	47.54	33.43	73.68	67.32	79.12	23.31	48.72
$\text{SGPT}_{5.8\text{B}}$ (Muennighoff, 2022)	50.25	56.56	40.34	82.00	68.13	78.10	31.46	58.93
$\text{GTR}_{\text{T5-XXL}}$ (Ni et al., 2022b)	48.48	56.65	42.42	86.12	67.41	78.38	30.64	58.97
Sentence-T5_{xxl} (Ni et al., 2022a)	42.24	56.42	43.72	85.07	73.42	82.63	30.08	59.51
$\text{UDEVER}_{\text{bloom-7b1}}$ (Zhang et al., 2023a)	49.34	55.91	40.81	85.40	72.13	83.01	30.97	60.63
$\text{Instructor}_{\text{xl}}$ (Su et al., 2023a)	49.26	57.29	44.74	86.62	73.12	83.06	**32.32**	61.79
$\text{BGE}_{\text{large-en-v1.5}}$ (Xiao et al., 2024)	54.29	60.03	46.08	87.12	75.97	83.11	31.61	64.23
$\text{UAE}_{\text{large-v1}}$ (Li and Li, 2024a)	54.66	59.88	46.73	87.25	75.58	84.54	32.03	64.64
Qwen2.5-7B								
EPIC (ours)	56.52	59.53	49.41	87.98	76.66	85.00	30.86	65.97
LLaMA-3.1-8B								
LLM2Vec (BehnamGhader et al., 2024)	56.63	59.68	46.45	87.80	75.92	83.58	30.94	65.01
Anchor (Su et al., 2025)	57.09	**61.38**	46.03	**88.92**	76.17	83.76	30.13	65.30
EPIC (ours)	57.08	59.22	48.67	87.98	77.03	85.38	31.26	66.10
Mistral-7B								
E5 (Wang et al., 2024a)	52.78	60.38	47.78	88.47	76.80	83.77	31.90	64.56
ECHO (Springer et al., 2025)	55.52	58.14	46.32	87.34	77.43	82.56	30.73	64.68
GRITLM (Muennighoff et al., 2024)	53.10	61.30	48.90	86.90	77.00	82.80	29.40	64.70
LLM2Vec (BehnamGhader et al., 2024)	55.99	58.42	45.54	87.99	76.63	84.09	29.96	64.80
Anchor(Su et al., 2025)	56.87	60.56	45.73	87.99	75.95	83.52	30.28	64.99
NV-Embed† (Lee et al., 2025a)	-	-	-	-	-	-	-	65.80
MGH (Pan et al., 2025)	57.49	58.80	47.96	87.83	**77.62**	84.04	31.10	65.87
bge-en-icl (Li et al., 2025)	**59.83**	56.83	46.78	88.54	77.51	84.08	30.39	66.18
EPIC (ours)	56.89	59.52	**49.56**	88.62	77.31	**85.49**	31.41	**66.37**

Table 1: Performance comparison on the full MTEB benchmark (56 datasets) among models trained exclusively on publicly available retrieval data. `Qwen2.5-7B`, `Mistral-7B`, and `LLaMA-3.1-8B` denote models built upon these LLMs, while `Miscellaneous` refers to methods using other base models. † represents the result is from Pan et al. (2025). The best result is highlighted in **bold**, and the second-best result is underlined.

Method	Qwen2.5-7B	LLaMA-3.1-8B	Mistral-7B
Baseline	65.05	65.25	65.33
$\text{EPIC}_{\text{w/o ICD}}$	$65.68_{+0.63}$	$65.89_{+0.64}$	$66.11_{+0.78}$
$\text{EPIC}_{\text{w/ ICD}}$	$\mathbf{65.97}_{+0.92}$	$\mathbf{66.10}_{+0.85}$	$\mathbf{66.37}_{+1.04}$

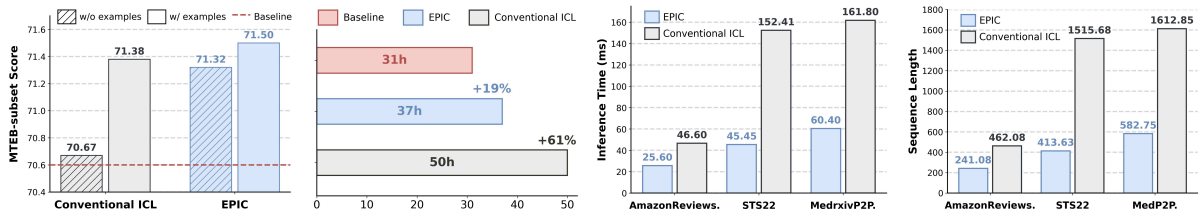
Table 2: Performance of **EPIC-trained** models with or without in-context demonstrations (ICD) during inference on MTEB (56 datasets). Baseline models are conventionally trained without any ICL strategy.

Evaluation. We verify the effectiveness of our method on the challenging Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), which consists of 56 datasets spanning 7 diverse embedding tasks. Given that evaluating a 7B-parameter model on MTEB requires hundreds of A100 GPU hours, we conduct ablations and analysis on a smaller 26-dataset subset of MTEB. For fair comparison, we construct fixed in-context prompts for each dataset based on the examples provided by bge-en-icl. More evaluation details are presented in Appendix B.

3.2 Main Results

Comparison to state-of-the-art methods. Since existing models (Lee et al., 2025b; Zhang et al., 2025; Zhao et al., 2025) often rely on extensive in-domain non-retrieval data from MTEB or proprietary synthetic datasets for training, it is difficult to ensure a fair academic comparison and reliably assess generalization to unseen tasks (Su et al., 2025; Li et al., 2025). To this end, we compare our EPIC only against models trained solely on publicly available retrieval datasets.

Table 1 presents the averaged scores for overall MTEB and its seven embedding task categories. Notably, our EPIC establishes new state-of-the-art performance across different LLM architectures. For the LLaMA-3.1-8B model, EPIC surpasses Anchor (66.13 vs. 65.30), which requires an additional full-parameter training stage before contrastive learning. For the widely adopted Mistral-7B model, EPIC achieves an average score of 66.37, outperforming E5 (64.56), ECHO (64.68), and bge-en-icl (66.18). Compared with bge-en-icl, which incorporates a conventional discrete ICL strategy, our findings suggest that embedding-based in-context prompting improves the representational capability more effectively. Moreover, EPIC exceeds competitive approaches that benefit from modified bidirectional attention on Mistral-



(a) Performance on MTEB subset (b) Training time (in hours) (c) Inference Speed (in milliseconds) (d) Rerquired Sequence Length

Figure 3: Comparison between EPIC and conventional ICL on Mistral-7B. (a) Performance comparison on the 26-dataset subset of MTEB with and without in-context examples during inference. (b) Training time on a single NVIDIA A100 80GB GPU. (c) Average inference time per sample on selected MTEB datasets (see Appendix C.4 for more details). (d) Average required sequence length on selected MTEB datasets.

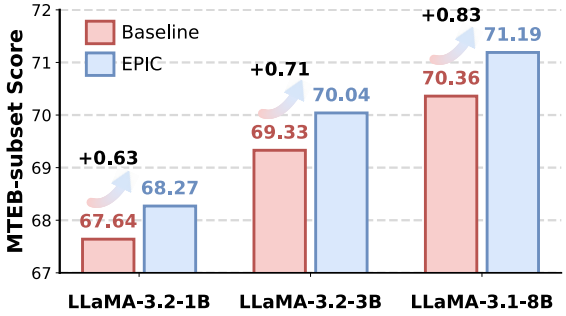


Figure 4: Performance comparison on the MTEB subset across different model scales, including LLaMA-3.2-1B, LLaMA-3.2-3B, and LLaMA-3.1-8B.

Sample-n	l	64	32	16
Average Score	**71.50**	71.40	71.32	71.30

Table 3: Performance of $\text{EPIC}_{\text{Mistral-7B}}$ on the MTEB subset by sampling $\frac{l}{n}$ tokens to represent the *query* or *passage* in demonstrations, where l denotes the sequence length of $\mathbf{Q}_i$ or $\mathbf{P}_i$, and $n \in \{l, 64, 32, 16\}$.

7B, including GritLM (64.70), LLM2Vec (64.80), NV-Embed (65.80), and MGH (65.87). These results consistently showcase the superior performance of our EPIC in enhancing LLMs as text encoders across diverse embedding tasks.

Comparison to the baselines. In Table 2, we compare our EPIC (w/ or w/o in-context demonstrations during inference) against standard contrastive learning baselines that do not incorporate any ICL strategy. Specifically, our method yields notable performance improvements of 0.92, 0.85, and 1.04 points over the baselines on Qwen2.5-7B, LLaMA-3.1-8B, and Mistral-7B, respectively. These results underscore the robustness and effectiveness of EPIC in improving embedding quality without relying on a specific base model.

Beyond improvements in in-context scenarios, we uncover an intriguing representational property: even without any in-context prompts at inference time, EPIC-trained models still achieve state-of-the-art performance, consistently outperforming baselines by 0.63, 0.64, and 0.78 points on Qwen2.5-7B, LLaMA-3.1-8B, and Mistral-7B, respectively. We attribute this to three key factors during training: (1) the random sampling strategy explicitly allows the model to work without demonstrations; (2) the demonstration embeddings are generated without reliance on in-context prompts; and (3) EPIC not only encourages the model to align semantically related embeddings, but also requires it to internalize the demonstration embeddings as part of the in-context prompt.

Furthermore, compared to the baselines, EPIC-trained models consistently reduce the proportion of attention assigned to the first token across different layers, thereby alleviating the attention sink phenomenon (Lin et al., 2025b). As a result, the EOS token is able to aggregate semantic information from the remaining tokens more effectively, leading to higher-quality embeddings. More details are provided in Appendix C.3.

Comparison to discrete ICL. To further examine the benefits of our method, we quantitatively compare it against the Mistral-7B model trained with conventional ICL under the same settings. As illustrated in Figure 3(a), our continuous embedding-based strategy matches the performance of discrete textual ICL while requiring a lower token budget. More importantly, the ICL counterpart fails to improve embedding quality when demonstrations are removed, underscoring the superiority of our method in non-ICL scenarios.

Moreover, as shown in Figure 3(b), conventional ICL increases training time by over 60% compared to the baseline, while EPIC incurs only about 19%

Method	Average Score
(a) In-Context Prompt Format	
w/o *query-passage*	70.71
w/o *passage*	71.17
w/o *query*	71.31
w/ only one instruction	71.30
(b) Compression Strategy	
Compress instruction and *query-passage*	71.22
Only compress *query*	71.35
Only compress *passage*	71.42
EPIC (ours)	**71.50**

Table 4: Performance comparison of $\text{EPIC}_{\text{Mistral-7B}}$ on the MTEB subset with different in-context prompt formats and compression strategies.

Method	Average Score
EPIC	**71.50**
w/ Learnable tokens	71.05
Soft-Prompt	70.83
Instruction-Tuning	70.60

Table 5: Performance comparison of $\text{EPIC}_{\text{Mistral-7B}}$ with other methods using learnable tokens on the MTEB subset, where Instruction-Tuning denotes the baseline trained using only task-specific instructions.

overhead by compressing discrete demonstrations into continuous vectors. In addition, Figure 3(c)-(d) confirm that EPIC consistently reduces token usage and yields lower inference latency on MTEB datasets, highlighting its efficiency in reducing computational cost during training and inference.

3.3 Ablation Studies

Robustness across models of different scales. Given the strong performance of EPIC on 7B and 8B models, we further evaluate its effectiveness at smaller scales. As shown in Figure 4, EPIC consistently improves the embedding capabilities of LLMs ranging from 1B to 8B parameters, showing its scalability across model sizes. Furthermore, we observe larger gains as model size increases, indicating the potential for EPIC to continuously benefit from more powerful LLMs.

The number of continuous vectors. By default, EPIC uses two text embeddings to replace the *query-passage* pair in discrete demonstrations. To examine whether using more continuous vectors could provide richer contextual information, we sample every n tokens from the LLM's output sequence and represent the *query* or *passage* with $\frac{l}{n}$ continuous vectors (referred to as sample-n), where l denotes the sequence length of $\mathbf{Q}_i$ or $\mathbf{P}_i$. Results for sample-$64/32/16$ are reported in Table 3. We observe that a single text embedding is sufficient for representing the *query* or *passage* in our setting, while increasing the number of continuous vectors does not yield performance improvements.

Impact of different in-context prompt formats. The in-context demonstration used in this work consists of a textual instruction followed by a *query-passage* embedding pair. To examine the importance of this prompt design, we investigate four alternative prompt formats: (1) using only the instruction without *query-passage* embeddings, where each $\mathbf{E}_i = [\mathbf{I}]$; (2) retaining the instruction and the *query* embedding while removing the *passage* embedding, i.e., $\mathbf{E}_i = [\mathbf{I}; \mathbf{q}_i]$; (3) discarding only the *query* embedding, i.e., $\mathbf{E}_i = [\mathbf{I}; \mathbf{p}_i]$; and (4) using only one instruction in the in-context prompt, yielding the input $[\mathbf{I}; \mathbf{q}_1; \mathbf{p}_1; \ldots; \mathbf{q}_k; \mathbf{p}_k; \mathbf{I}; \mathbf{X}; [\texttt{EOS}]]$. The results in Table 4(a) indicate that all these variants lead to performance degradation, confirming the necessity of preserving the complete in-context prompt format adopted by EPIC.

Impact of different compression strategies. In conventional ICL for embedding tasks (Li et al., 2025), each textual demonstration consists of an instruction and a *query-passage* pair. To challenge this paradigm, EPIC compresses both the discrete *query* and *passage* into their corresponding continuous embeddings. We further evaluate three alternative compression strategies: (1) transforming both the instruction and the *query-passage* pair into text embeddings; (2) compressing only the *query*, and (3) compressing only the *passage*. As demonstrated in Table 4(b), EPIC exhibits the best trade-off between embedding performance and token usage. We hypothesize that jointly compressing the *query-passage* pair during training encourages the model to better understand and utilize its generated embeddings, while retaining the textual instruction effectively promotes the ICL capability.

Comparison with soft-prompt. Since both soft prompts and our method fundamentally leverage continuous vectors to encode semantic information instead of hard prompts, we compare EPIC with two alternative setups to further highlight our contributions: (1) replacing the demonstration embeddings in EPIC with the same number of learnable tokens, and (2) following common practices (Lester

Method	Average Score
EPIC	**71.50**
w/ Bi. + EOS pooling	70.83
w/ Bi. + Mean pooling	70.93
w/ Bi. + NV-Embed pooling	71.08

Table 6: Performance of $\text{EPIC}_{\text{Mistral-7B}}$ on the MTEB subset using bidirectional (Bi.) attention with various pooling strategies. Note: EPIC preserves the original causal attention and employs EOS pooling by default.

et al., 2021; Li and Liang, 2021) by prepending a set of learnable tokens as soft prompts to the input. All experiments are optimized with LoRA. The results in Table 5 show that EPIC achieves the best results, indicating that our embedding-based strategy provides richer semantic information in the continuous space than learnable tokens.

Influence of various attention and pooling mechanism. Recent studies achieve strong text embeddings by transforming the model's attention from causal to bidirectional (Li and Li, 2024b; Muennighoff et al., 2024; BehnamGhader et al., 2024; Lee et al., 2025a; Pan et al., 2025). To investigate the potential of this paradigm in our framework, we evaluate EPIC under bidirectional attention with various pooling strategies, including last-token pooling, mean pooling, and NV-Embed pooling (Lee et al., 2025a). As shown in Table 6, we observe that switching to bidirectional attention considerably degrades EPIC's performance, regardless of the pooling mechanism, consistent with previous findings (Li et al., 2025; Lin et al., 2025a). We speculate that the attention mismatch between pre-training and fine-tuning disrupts the advanced instruction-following capabilities of LLMs when provided with in-context demonstrations.

4 Related Work

Text Embeddings. Text embeddings are continuous vector representations that encode the contextual semantics of natural language text, facilitating a wide range of natural language language processing (NLP) tasks such as text classification (Logeswaran and Lee, 2018), question answering (Karpukhin et al., 2020a), and information retrieval (IR) (Jiang et al., 2026). Early efforts focused on word-level embeddings (Mikolov et al., 2013; Pennington et al., 2014), while later attempts learned fixed-length representations for variable-length texts by combining word vectors (Wieting et al., 2015; Wang et al., 2016). Modern approaches predominantly rely on pre-trained language models, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and T5 (Raffel et al., 2020) to generate contextualized text embeddings. Notable methods in this paradigm include SBERT (Reimers and Gurevych, 2019), SimCSE (Gao et al., 2021), and Sentence-T5 (Ni et al., 2022a), which are fine-tuned on natural language inference datasets. To further improve embedding performance, advanced techniques such as E5 (Wang et al., 2022), GTE (Li et al., 2023), and BGE (Xiao et al., 2024) employ weakly supervised contrastive learning on large-scale text pair corpora curated from web sources. More recent work attempts to develop general-purpose embedding models tailored to diverse tasks and domains through well-designed instruction-tuning (Su et al., 2023b; Wang et al., 2024b).

LLM-based Text Embedding. With the rapid advancement of large language models (LLMs), substantial efforts have been devoted to adapting them into strong embedding models. RepLLaMA (Ma et al., 2024) and LLaMA2Vec (Liu et al., 2024a) show that fine-tuning LLaMA-2-7B (Touvron et al., 2023) substantially improves the performances on retrieval tasks. To further obtain high-quality text embeddings, Wang et al. (2024a) fine-tune Mistral-7B (Jiang et al., 2023) on diverse synthetic data with standard contrastive loss, achieving competitive results. ECHO (Springer et al., 2025) repeats the input twice and extracts embeddings from the repeated sequence. Anchor (Su et al., 2025) enhances the semantic capacity of the EOS token by introducing an additional training stage before contrastive learning. As the first work to enable bidirectional attention in LLMs for embedding generation, BeLLM (Li and Li, 2024b) removes the causal mask at specific attention layers. Building on this foundation, many subsequent methods modify the LLMs to be fully bidirectional, including GRITLM (Muennighoff et al., 2024) and LLM2Vec (BehnamGhader et al., 2024), while NV-Embed (Lee et al., 2025a) and MGH (Pan et al., 2025) further propose novel pooling strategies to overcome the limitation of mean pooling. In addition, PromptEOL (Jiang et al., 2024) and bge-en-icl (Li et al., 2025) incorporate task-related demonstrations into the input to activate the in-context learning capabilities (Brown et al., 2020) of LLMs. In this work, we aim to enhance LLMs as embedding models by leverag-

ing ICL while mitigating its significant token cost through compressing discrete textual demonstrations into continuous embeddings.

Vector-based ICL. In-context learning (ICL) has become a powerful learning paradigm for LLMs, yet its underlying mechanisms remain unclear. Hendel et al. (2023) show that ICL operates by compressing a training set into a single task vector that guides the model to generate desired outputs. Building on this perspective, Yang et al. (2025) investigate potential factors in the emergence of task vectors. Moreover, Zhuang et al. (2025) demonstrate that pre-training projection modules with language modeling objectives enable effective vector-based ICL. Notably, these methods are developed for generative tasks. In contrast, to the best of our knowledge, this work presents the first embedding model that replaces discrete ICL demonstrations with their corresponding text embeddings, thus improving the representational capability of LLMs.

5 Conclusion

In this work, we introduced a novel embedding-based in-context prompt training strategy to improve the embedding capabilities of LLMs. Our method replaces conventional discrete demonstrations with their continuous embeddings, allowing the model to benefit from ICL while effectively reducing token overhead. Extensive experiments on MTEB demonstrated that EPIC achieves new state-of-the art results among models trained solely on publicly available retrieval datasets. Moreover, EPIC-enhanced models exhibited strong embedding performance even without any in-context prompt, further confirming the effectiveness and practicality of our method. We hope this work provides new perspective on prompting strategies for advancing the representation learning of LLMs.

Limitations

Despite the strong embedding results achieved by EPIC, there remain several limitations that need to be acknowledged: (1) Models that perform exceptionally well on MTEB, such as Qwen3-Embedding (Zhang et al., 2025), Gemini Embedding (Lee et al., 2025b), and KaLM-Embedding (Zhao et al., 2025), typically rely on extensive synthetic or MTEB-related data during training. Incorporating such training corpora could help further validate the effectiveness and generalizability of our approach. (2) Due to hardware constraints, we evaluate the proposed method only on LLMs ranging from 1B to 8B parameters, which also ensures fair comparison with prior work (Su et al., 2025; Pan et al., 2025; Lee et al., 2025a; Li et al., 2025). Scaling the experiments to larger model sizes, such as 30B or 70B, would make this work more comprehensive and meaningful. (3) Although this work provides new perspectives on embedding prompting, the underlying mechanisms of ICL for embedding generation remain unclear. Future work aims to provide a mechanistic explanation of ICL and further exploit its potential for text embedding.

Ethical Considerations

This work focuses on improving LLMs as text encoders, enabling a wide range of real-world applications such as information retrieval, question answering, and recommendation systems. However, it should be noted that our method may inherit and potentially amplify social biases (Hida et al., 2025) and hallucination issues (Bang et al., 2025) inherent in LLMs. Therefore, users are encouraged to apply our research in an ethical and responsible manner. In addition, we rely solely on publicly available datasets for training and open-source benchmarks for evaluation, both of which have been widely adopted in academic research, helping to mitigate ethical concerns to a certain extent.

References

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. HalluLens: LLM hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24128–24156.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The*

Twelfth International Conference on Learning Representations.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. 2017. Quora question pairs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46. Association for Computational Linguistics.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333.

Rem Hida, Masahiro Kaneko, and Naoaki Okazaki. 2025. Social bias evaluation for large language models requires prompt variations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14507–14530.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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*.

Angqing Jiang, Jianlyu Chen, Zhe Fang, Yongcan Wang, Xinpeng Li, Keyu Ding, and Defu Lian. 2026. Cmedteb & care: Benchmarking and enabling efficient chinese medical retrieval via asymmetric encoders. *arXiv preprint arXiv:2604.10937*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. 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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. 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.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025a. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. 2025b. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Kiela Douwe. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu.

2025. Making text embedders few-shot learners. In *The Thirteenth International Conference on Learning Representations*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Xianming Li and Jing Li. 2024a. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839.

Xianming Li and Jing Li. 2024b. BeLLM: Backward dependency enhanced large language model for sentence embeddings. 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 792–804.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Ailiang Lin, Zhuoyun Li, Kotaro Funakoshi, and Manabu Okumura. 2025a. Causal2vec: Improving decoder-only llms as versatile embedding models. *arXiv preprint arXiv:2507.23386*.

Ziyong Lin, Haoyi Wu, Shu Wang, Kewei Tu, Zilong Zheng, and Zixia Jia. 2025b. Look both ways and no sink: Converting llms into text encoders without training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22839–22853.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zheng Liu, Chaofan Li, Shitao Xiao, Yingxia Shao, and Defu Lian. 2024a. Llama2Vec: Unsupervised adaptation of large language models for dense retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3490–3500.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. ChatQA: Surpassing GPT-4 on conversational QA and RAG. In *Advances in Neural Information Processing Systems*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

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*, pages 2421–2425.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A human-generated MAchine reading COmprehension dataset.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022b. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

Tengyu Pan, Zhichao Duan, Zhenyu Li, Bowen Dong, Ning Liu, Xiuxing Li, and Jianyong Wang. 2025. Negative matters: Multi-granularity hard-negative synthesis and anchor-token-aware pooling for enhanced text embeddings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31102–31118.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou,

Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2025. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*.

Chang Su, Dengliang Shi, Siyuan Huang, Jintao Du, Changhua Meng, Yu Cheng, Weiqiang Wang, and Zhouhan Lin. 2025. Training llms to be better text embedders through bidirectional reconstruction. *arXiv preprint arXiv:2509.03020*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023a. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023b. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 11897–11916.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Yashen Wang, Heyan Huang, Chong Feng, Qiang Zhou, Jiahui Gu, and Xiong Gao. 2016. CSE: Conceptual sentence embeddings based on attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 505–515.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, and Jin Ma. 2023. T2ranking: A large-scale chinese benchmark for passage ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2681–2690.

Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, and Robert Nowak. 2025. Task vectors in in-context learning: Emergence, formation, and benefit. *arXiv preprint arXiv:2501.09240*.

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

Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. 2023a. Language models are universal embedders. *arXiv preprint arXiv:2310.08232*.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023b. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, et al. 2025. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model. *arXiv preprint arXiv:2506.20923*.

Yufan Zhuang, Chandan Singh, Liyuan Liu, Jingbo Shang, and Jianfeng Gao. 2024. Vector-icl: In-context learning with continuous vector representations. *arXiv preprint arXiv:2410.05629*.

Yufan Zhuang, Chandan Singh, Liyuan Liu, Jingbo Shang, and Jianfeng Gao. 2025. Vector-ICL: In-context learning with continuous vector representations. In *The Thirteenth International Conference on Learning Representations*.

A Experimental Details for Training

A.1 Training Setup

In this section, we provide additional training details based on Section 3.1. We fine-tune Mistral-7B for 1000 steps, and Qwen2.5-7B as well as LLaMA-3.1-8B for 800 steps. We adopt the AdamW optimizer with 300 warm-up steps, followed by a linear learning-rate decay over the remaining steps. To ensure fair comparison, we follow the open-source implementation of LLM2Vec (BehnamGhader et al., 2024) and set the random seed to 42 across all experiments. To reduce GPU memory usage, we enable bfloat16 precision, FlashAttention-2 (Dao, 2024), and gradient checkpointing. Following Pan et al. (2025), we further employ gradient accumulation of 8 to simulate a batch size of 512. Additionally, we ensure that all samples within each batch are drawn from the same dataset

A.2 Public Retrieval Datasets

Following Springer et al. (2025), the collection of publicly available retrieval datasets used for training is distributed under the Apache License 2.0 and includes the following datasets: ELI5 (sample ratio 0.1) (Fan et al., 2019), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), MIRACL (Zhang et al., 2023b), MS-MARCO passage ranking (sample ratio 0.5) and document ranking (sample ratio 0.2) (Nguyen et al., 2017), NQ (Karpukhin et al., 2020b), NLI (Gao et al., 2021), SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), Quora Duplicate Questions (sample ratio 0.1) (DataCanary et al., 2017), Mr. TyDi (Zhang et al., 2021), DuReader (He et al., 2018), and T2Ranking (sample ratio 0.5) (Xie et al., 2023).

Following (BehnamGhader et al., 2024), we use different instructions for each retrieval dataset during training, as listed in Table 7. It is worth noting that for `query`–`passage` sample pairs, we apply instructions only to the `query`, while leaving the `passage` unchanged.

B Experimental Details for Evaluation

B.1 Massive Text Embeddings Benchmark (MTEB)

In line with previous work (Wang et al., 2024a; BehnamGhader et al., 2024; Springer et al., 2025; Lee et al., 2025a; Su et al., 2025; Pan et al., 2025; Li et al., 2025), we adopt the large-scale MTEB English subsets (Muennighoff et al., 2023) to evaluate the effectiveness of our method. This benchmark is distributed under the Apache License 2.0 and comprises 56 English datasets across seven diverse embedding task categories: retrieval (Retr.), reranking (Rerank.), clustering (Clust.), pair classification (PairClass.), classification (Class.), semantic textual similarity (STS), and summarization (Summ.). The corresponding evaluation metrics are nDCG@10, MAP, V-measure (V-meas.), average precision (AP), accuracy (Acc.), and Spearman correlation (Spear., both for STS and Summ.), respectively.

For fair comparison, we directly employ the in-context demonstrations curated by bge-en-icl (Li et al., 2025), which provide between one and eight sentence pairs for each MTEB dataset. Since these examples are specifically selected for bge-en-icl, they may be suboptimal for our method. Therefore, for datasets where the demonstrations fail to improve performance, we simply disable in-context prompting. Notably, for asymmetric tasks such as retrieval, instructions or in-context prompts are applied only to the `query`, whereas for symmetric tasks, they are applied to both input texts. The instructions used for each MTEB dataset are listed in

Dataset	Instruction (s)
ELI5	Provided a user question, retrieve the highest voted answers on Reddit ELI5 forum
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
FEVER	Given a claim, retrieve documents that support or refute the claim
MIRACL	Given a question, retrieve Wikipedia passages that answer the question
MSMARCO Passage	Given a web search query, retrieve relevant passages that answer the query
MSMARCO Document	Given a web search query, retrieve relevant documents that answer the query
NQ	Given a question, retrieve Wikipedia passages that answer the question
NLI	Given a premise, retrieve a hypothesis that is entailed by the premise Retrieve semantically similar text
SQuAD	Retrieve Wikipedia passages that answer the question
TriviaQA	Retrieve Wikipedia passages that answer the question
QuoraDuplicates	Given a question, retrieve questions that are semantically equivalent to the given question Find questions that have the same meaning as the input question
Mr. TyDi	Given a question, retrieve Wikipedia passages that answer the question
DuReader	Given a Chinese search query, retrieve web passages that answer the question
T2Ranking	Given a Chinese search query, retrieve web passages that answer the question

Table 7: Instructions used for publicly available retrieval datasets during training.

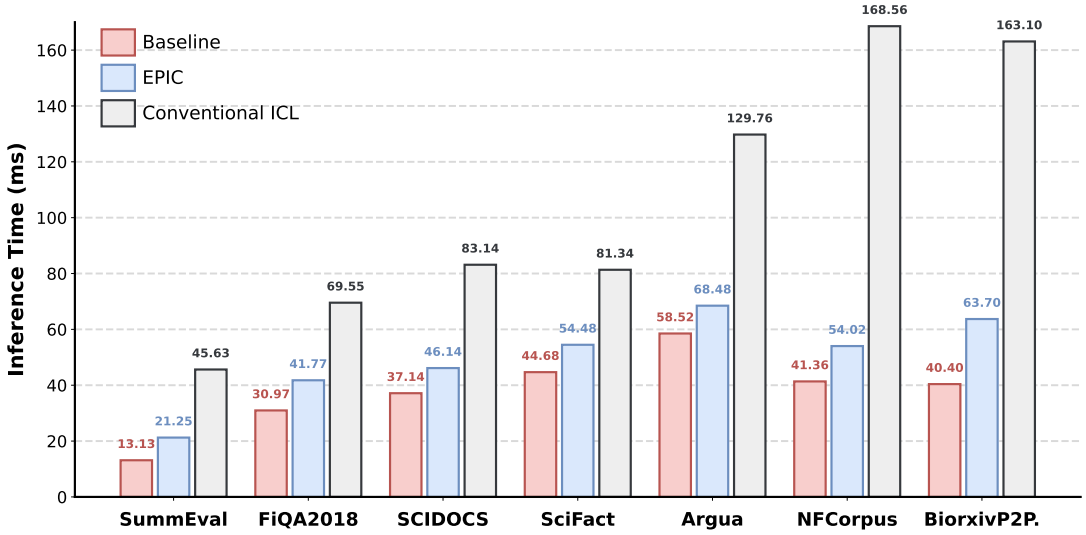


Figure 5: Average per-sample inference latency of Mistral-7B–based methods on selected MTEB datasets. The baseline refers to the standard Mistral-7B model with EOS pooling. All results are obtained with a batch size of 64 on a single NVIDIA A100 GPU. For asymmetric retrieval datasets, latency is reported per `query`–`passage` pair.

Table 11.

B.2 MTEB Subset

The full MTEB benchmark contains over ten millions samples and requires hundreds of A100-80GB GPU hours to evaluate a 7B-parameter model. To accelerate ablation studies and analysis, we follow MGH (Pan et al., 2025) and select a representative subset of MTEB comprising 26 datasets: BIOSSES, STS12, STS13, STS14, STS15, STS16, STS17, STS22, STS-Benchmark, SICK-R, AmazonReviewsClassification, MTOPDomainClassification, TweetSentimentExtractionClassification, ImdbClassification, TwitterSemEval2015, TwitterURLCorpus, SciFact, NFCorpus, FiQA2018, SCIDOCS, BiorxivClusteringS2S, MedrxivClusteringS2S, TwentyNewsgroupsClustering, AskUbuntuDupQuestions, StackOverflowDupQuestions, and SciDocsRR.

C Additional Results

C.1 Inference Latency

In this section, we further report the inference latency of our EPIC on MTEB datasets. As shown in Figure 5, by compressing discrete textual demonstrations into embedding-based continuous representations, EPIC reduces inference time by up to 70% compared with conventional ICL (e.g., STS22: 45.45 vs. 152.41). There findings demonstrate that

# of examples	Average Score
0	70.60
1	71.15
2	71.30
5	**71.50**
8	71.43

Table 8: Performance comparison of $\text{EPIC}_{\text{Mistral-7B}}$ with varying numbers of in-context demonstrations during fine-tuning on the MTEB subset, where 0 examples refers to training without any ICL strategy.

our approach substantially mitigates the token burden during inference.

C.2 The Number of In-Context Examples During Training

By default, we randomly sample five demonstrations from the same batch during fine-tuning, following bge-en-icl (Li et al., 2025). We further investigate the impact of using 1, 2, and 8 demonstrations. As shown in Table 8, compared to the baseline model trained without any ICL strategy, using even a single demonstration during training leads to performance improvements. However, when the number of demonstrations is increased from five to eight, the embedding performance no longer improves, while the training cost becomes higher. Overall, to ensure a fair comparison with prior work and to strike a balance between performance and computational efficiency, we use five in-context demonstrations during training in this work.

C.3 Analysis of the Attention Sink Phenomenon

The attention sink phenomenon refers to the model's tendency to focus excessively on the first token, which has been shown to hinder the performance of embedding models (Lin et al., 2025b). We conduct an attention analysis on $\text{EPIC}_{\text{Mistral-7B}}$ by computing the average proportion of attention that the EOS token assigns to the first token across different layers. As shown in Table 9, EPIC-trained models consistently alleviate the attention sink phenomenon both with and without in-context demonstrations during inference. Consequently, the EOS token, which serves as the output text embedding, can attend more effectively to the remaining tokens, thereby improving the embedding quality.

Method	Layer 8	Layer 16	Layer 24	Layer 32
Baseline	**0.75**	**0.58**	**0.70**	**0.46**
$\text{EPIC}_{\text{w/o ICD}}$	0.57	0.40	0.47	0.33
$\text{EPIC}_{\text{w/ ICD}}$	0.54	0.37	0.45	0.31

Table 9: Proportion of attention assigned to the first token across selected layers for EPIC-trained models with or without in-context demonstrations (ICD) during inference. Baseline models are conventionally trained without any ICL strategy.

LoRA Rank	Average Score
16	71.39
32	71.42
64	**71.50**

Table 10: Performance comparison of $\text{EPIC}_{\text{Mistral-7B}}$ with different LoRA ranks on the MTEB subset.

C.4 The Impact of LoRA Rank

In addition to the LoRA rank of 64 used in our experiments, we further examine the model performance with LoRA ranks of 16 and 32. As presented in Table 10, EPIC achieves strong results even with smaller LoRA rank, demonstrating its robustness across different LoRA settings. For a fair comparison with the previous state-of-the-art method, bge-en-icl, we adopt a LoRA rank of 64 as the default setting in this work.

C.5 Detailed MTEB Results

We present the detailed results of the proposed EPIC on all MTEB datasets using three base models: Qwen2.5-7B, Mistral-7B, and LLaMA-3.1-8B, as summarized in Table 12.

Dataset	Instruction
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual.
AmazonPolarityClassification	Classify Amazon reviews into positive or negative sentiment
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category
Banking77Classification	Given a online banking query, find the corresponding intents
EmotionClassification	Classify the emotion expressed in the given Twitter message into one of the six emotions: anger, fear, joy, love, sadness, and surprise.
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset.
MassiveIntentClassification	Given a user utterance as query, find the user intents
MassiveScenarioClassification	Given a user utterance as query, find the user scenarios
MTOPDomainClassification	Classify the intent domain of the given utterance in task-oriented conversation
MTOPIntentClassification	Classify the intent of the given utterance in task-oriented conversation
ToxicConversationsClassif.	Classify the given comments as either toxic or not toxic
TweetSentimentClassification	Classify the sentiment of a given tweet as either positive, negative, or neutral
ArxivClusteringP2P	Identify the main and secondary category of Arxiv papers based on the titles and abstracts.
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles
RedditClustering	Identify the topic or theme of Reddit posts based on the titles
RedditClusteringP2P	Identify the topic or theme of Reddit posts based on the titles and posts
StackExchangeClustering	Identify the topic or theme of StackExchange posts based on the titles
StackExchangeClusteringP2P	Identify the topic or theme of StackExchange posts based on the given paragraphs
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles
SprintDuplicateQuestions	Retrieve duplicate questions from Sprint forum
TwitterSemEval2015	Retrieve tweets that are semantically similar to the given tweet
TwitterURLCorpus	Retrieve tweets that are semantically similar to the given tweet
AskUbuntuDupQuestions	Retrieve duplicate questions from AskUbuntu forum
MindSmallReranking	Retrieve relevant news articles based on user browsing history
SciDocsRR	Given a title of a scientific paper, retrieve the titles of other relevant papers
StackOverflowDupQuestions	Retrieve duplicate questions from StackOverflow forum
ArguAna	Given a claim, find documents that refute the claim
ClimateFEVER	Given a claim about climate change, retrieve documents that support or refute the claim.
CQADupstackRetrieval	Given a question, retrieve detailed question descriptions from Stackexchange that are duplicates to the given question.
DBPedia	Given a query, retrieve relevant entity descriptions from DBPedia
FEVER	Given a claim, retrieve documents that support or refute the claim
FiQA2018	Given a financial question, retrieve user replies that best answer the question
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
MSMARCO	Given a web search query, retrieve relevant passages that answer the query
NFCorpus	Given a question, retrieve relevant documents that best answer the question
NQ	Given a question, retrieve Wikipedia passages that answer the question
QuoraRetrieval	Given a question, retrieve questions that are semantically equivalent to the given question.
SCIDOCS	Given a scientific paper title, retrieve paper abstracts that are cited by the given paper
SciFact	Given a scientific claim, retrieve documents that support or refute the claim
Touche2020	Given a question, retrieve detailed and persuasive arguments that answer the question
TRECCOVID	Given a query on COVID-19, retrieve documents that answer the query
STS*	Retrieve semantically similar text.
SummEval	Given a news summary, retrieve other semantically similar summaries

Table 11: Instructions used for MTEB evaluation. "STS*" indicates that the instruction is applied to all STS datasets.

Dataset	Qwen2.5-7B	Mistral-7B	LLaMA-3.1-8B
AmazonCounterfactualClassification	70.55	77.54	76.87
AmazonPolarityClassification	96.13	95.76	94.76
AmazonReviewsClassification	54.56	53.91	51.58
ArguAna	61.37	60.65	61.76
ArxivClusteringP2P	51.09	49.69	49.14
ArxivClusteringS2S	47.71	46.28	45.74
AskUbuntuDupQuestions	64.55	65.96	64.86
BIOSSES	86.45	85.99	86.12
Banking77Classification	87.88	88.71	88.59
BiorxivClusteringP2P	41.19	39.41	40.67
BiorxivClusteringS2S	38.66	38.04	38.52
CQADupstackRetrieval	47.13	44.84	46.95
ClimateFEVER	35.12	34.01	35.48
DBPedia	49.81	50.77	50.60
EmotionClassification	51.02	50.49	49.48
FEVER	90.55	91.47	90.90
FiQA2018	51.91	54.64	53.09
HotpotQA	67.78	73.06	72.72
ImdbClassification	93.74	92.26	92.80
MSMARCO	40.82	42.04	42.13
MTOPDomainClassification	96.43	96.04	96.32
MTOPIntentClassification	82.90	84.18	85.94
MassiveIntentClassification	79.87	79.23	78.95
MassiveScenarioClassification	81.95	82.10	81.30
MedrxivClusteringP2P	35.70	35.13	33.77
MedrxivClusteringS2S	34.51	34.92	32.30
MindSmallReranking	33.14	32.48	32.42
NFCorpus	40.68	41.07	40.59
NQ	63.83	65.81	67.18
QuoraRetrieval	89.27	89.38	89.24
RedditClustering	61.27	65.26	64.36
RedditClusteringP2P	63.48	66.96	65.03
SCIDOCS	23.84	22.03	22.87
SICK-R	83.57	83.80	84.10
STS12	78.74	79.04	79.50
STS13	89.17	89.18	89.42
STS14	84.81	85.72	85.77
STS15	89.61	90.59	90.04
STS16	87.99	88.44	88.33
STS17	91.86	92.78	91.95
STS22	69.20	69.99	69.39
STSBenchmark	88.69	89.38	89.21
SciDocsRR	86.61	84.42	85.76
SciFact	77.37	76.90	77.66
SprintDuplicateQuestions	96.73	96.58	96.96
StackExchangeClustering	74.99	73.71	71.65
StackExchangeClusteringP2P	39.38	38.68	36.38
StackOverflowDupQuestions	53.81	55.21	53.82
SummEval	30.86	31.41	31.26
TRECCOVID	83.60	83.10	79.65
Touche2020	24.70	23.56	25.32
ToxicConversationsClassification	61.34	65.20	65.55
TweetSentimentExtractionClassification	63.58	62.25	62.17
TwentyNewsgroupsClustering	55.51	57.12	57.81
TwitterSemEval2015	79.94	82.13	80.02
TwitterURLCorpus	87.28	87.16	86.97
MTEB Average (56)	65.97	66.37	66.10

Table 12: Results of EPIC on each MTEB datasets across three base models: Qwen2.5-7B, Mistral-7B, and LLaMA-3.1-8B.