Training LLMs to be Better Text Embedders through Bidirectional Reconstruction

Chang Su[1], **Dengliang Shi**[2], **Siyuan Huang**[1], **Jintao Du**[2], **Changhua Meng**[2],
Yu Cheng[2], **Weiqiang Wang**[2], **Zhouhan Lin**[1,*]
[1]LUMIA Lab, Shanghai Jiao Tong University
[2]Tiansuan Lab, Ant Group Co., Ltd.
suchang0912@sjtu.edu.cn, siyuan_huang_sjtu@outlook.com, lin.zhouhan@gmail.com
{dengliang.sdl,lingke.djt,changhua.mch,cy122623,weiqiang.wwq}@antgroup.com

Abstract

Large language models (LLMs) have increasingly been explored as powerful text embedders. Existing LLM-based text embedding approaches often leverage the embedding of the final token, typically a reserved special token such as [EOS]. However, these tokens have not been intentionally trained to capture the semantics of the whole context, limiting their capacity as text embeddings, especially for retrieval and re-ranking tasks. We propose to add a new training stage before contrastive learning to enrich the semantics of the final token embedding. This stage employs bidirectional generative reconstruction tasks, namely EBQ2D (Embedding-Based Query-to-Document) and EBD2Q (Embedding-Based Document-to-Query), which interleave to anchor the [EOS] embedding and reconstruct either side of Query-Document pairs. Experimental results demonstrate that our additional training stage significantly improves LLM performance on the Massive Text Embedding Benchmark (MTEB), achieving new state-of-the-art results across different LLM base models and scales.[1]

1 Introduction

Text embeddings serve as the foundation for many natural language processing (NLP) tasks by capturing the semantic meaning of text in vector representations (Muennighoff et al., 2023; Lewis et al., 2020). For example, in text retrieval, both queries and documents are encoded into a shared latent space, where their relevance is measured by embedding similarity, which in turn places strong demands on embedding quality (Karpukhin et al., 2020).

Early studies leveraged pre-trained language models with bidirectional attention, such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), to generate high-quality text embeddings. These approaches typically relied on complex multi-stage training and large-scale annotated pairs (Wang et al., 2022; Xiao et al., 2024).

More recently, the impressive semantic understanding capability of large language models (LLMs) has attracted growing interest in their use for embedding tasks. Some approaches transform LLMs into text encoders by enabling bidirectional attention (BehnamGhader et al., 2024; Muennighoff et al., 2024), but such architectural modifications compromise the unification between generation and embedding. Alternatively, other methods retain the auto-regressive nature and causal attention, deriving embeddings from the final token (e.g., <\s> or [EOS]) to capture global context, a practice that has been widely adopted (Li et al., 2025; Springer et al., 2025). However, during general pre-training, these tokens serve merely as sequence delimiters, and the model does not learn to encode contextual semantics into their representations or to establish meaningful alignment between relevant texts with them. This greatly limits the potential of LLMs in embedding tasks.

Motivated by this, we propose a new training stage before contrastive learning, establishing a two-stage training framework as illustrated in Figure 1. For the new stage, we introduce two bidirectional reconstruction tasks, **EBQ2D** (Embedding-Based Query-to-Document) and **EBD2Q** (Embedding-Based Document-to-Query), which treat the [EOS] embedding as an anchor to: 1) aggregate the semantic information of either the query or the relevant document, and 2) serve as the reference for generating its counterpart. Specifically, in the EBQ2D task, the output embedding of the [EOS] token in the query is used to prompt the model to generate the relevant document. This encourages the model to embed the

[*] Corresponding Author.

[1] Our code is available at https://github.com/LUMIA-Group/Anchor-Embedding.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 4351–4369
November 4-9, 2025 ©2025 Association for Computational Linguistics

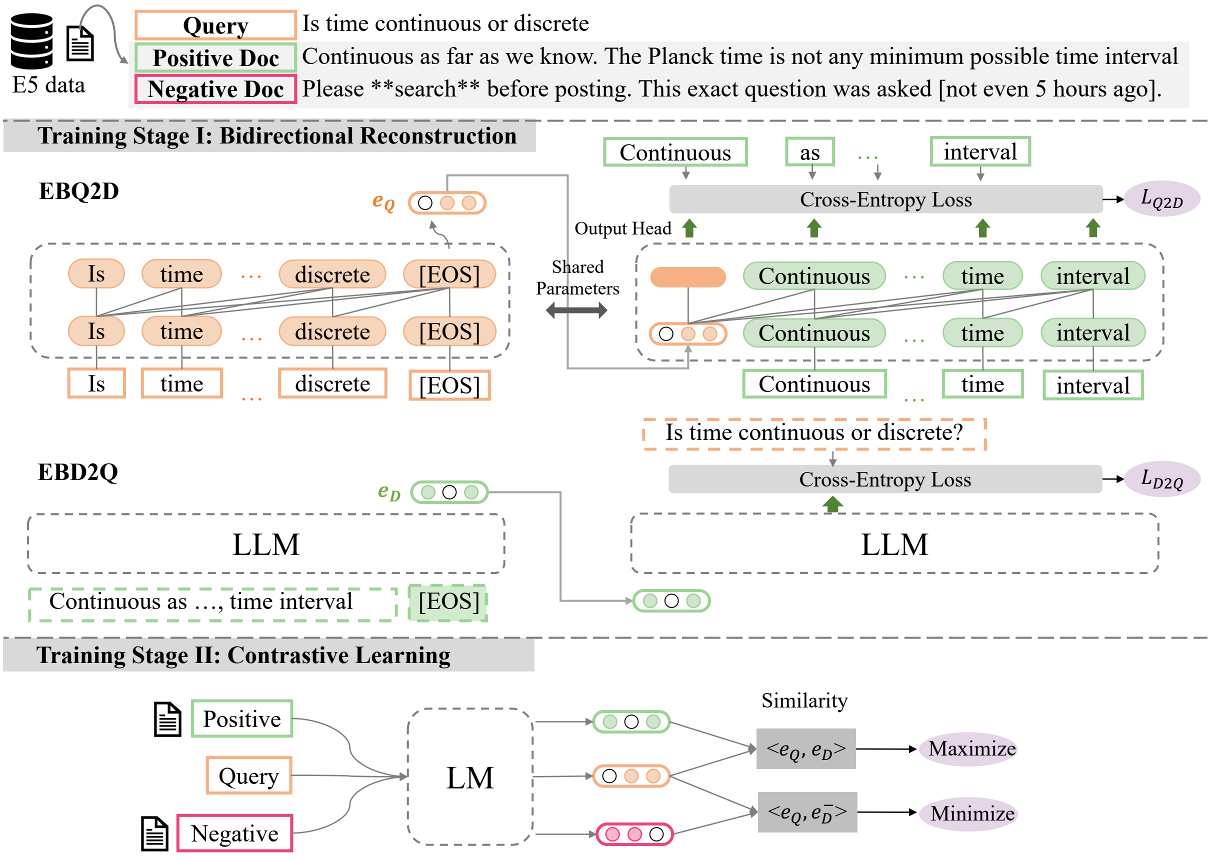

E5 data
Query
Is time continuous or discrete
Positive Doc
Continuous as far as we know. The Planck time is not any minimum possible time interval
Negative Doc
Please **search** before posting. This exact question was asked [not even 5 hours ago].
Training Stage I: Bidirectional Reconstruction
EBQ2D
e_Q
Is
time
…
discrete
[EOS]
Continuous
as
…
interval
Cross-Entropy Loss
L_{Q2D}
Output Head
Shared Parameters
Continuous
…
time
interval
Is time continuous or discrete?
Cross-Entropy Loss
L_{D2Q}
EBD2Q
e_D
LLM
Continuous as …, time interval
[EOS]
LLM
Training Stage II: Contrastive Learning
Positive
Query
Negative
LM
Similarity
$<e_Q, e_D>$
Maximize
$<e_Q, e_D^->$
Minimize

Figure 1: The pipeline of our approach. The model is first trained using two bidirectional reconstruction tasks, followed by contrastive learning. The public E5 data serves as the training corpus for both stages.

semantics of the query and, more importantly, the implied document-level content within the embedding. Symmetrically, the EBD2Q task uses the [EOS] embedding of the document to guide the generation of the corresponding query, training the model to reason backward from content to intent. This training stage enhances the model's ability to capture implicit semantic relationships between queries and relevant documents via the output embedding. Following the bidirectional reconstruction, the second stage fine-tunes the model with contrastive learning to further improve the quality of the generated representations.

We apply our framework to several decoder-only LLMs, including LLaMA-3.1, LLaMA-3.2, Qwen2.5, and Mistral, with their sizes varying from 1B to 8B. Experimental results demonstrate that our proposed bidirectional reconstruction training consistently improves performance across different models and scales. Notably, our method achieves new state-of-the-art results on the Massive Text Embeddings Benchmark (MTEB) (Muennighoff et al., 2023) among models trained solely on publicly available data. Meanwhile, comprehensive ablation studies further validate the effectiveness of our proposed training objectives and the two-stage framework.

In summary, our key contributions are as follows:

- We highlight a mismatch in the role of the [EOS] token between general language model pre-training and embedding tasks.
- We introduce a novel training stage consisting of two bidirectional generative reconstruction tasks, EBQ2D and EBD2Q, that encourage the model to inject semantic alignment into the [EOS] representation.
- Our approach consistently improves the quality of embeddings generated by LLMs, achieving new state-of-the-art results on MTEB.

2 Related Works

Text embeddings. Text embeddings are vector representations of natural language text that encode its semantic content, which play a pivotal role in various natural language processing (NLP) tasks, such as information retrieval (IR), semantic similarity estimation, classification, and clustering (Fujiwara et al., 2023; Karpukhin et al., 2020). As an example, the first-stage retrieval in an IR system leverages embedding similarity to retrieve relevant documents from a large-scale corpus. Apart from early attempts using latent semantic indexing (Deerwester et al., 1990) and word-level representations (Mikolov et al., 2013), modern research on embedding task utilizes pre-trained language models, like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2020), and T5 (Raffel et al., 2020), significantly outperforming traditional approaches. To further enhance the performance, advanced methods like E5 (Wang et al., 2022) and BGE (Xiao et al., 2024) employ a complex multi-stage training pipeline consisting of large-scale weakly supervised contrastive pre-training and multi-task fine-tuning. More recently, LLMs have become the new foundation for text embedding given their superior capability on semantic understanding (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). LLM2Vec (BehnamGhader et al., 2024) enables bidirectional attention and applies masked language modeling to transform decoder-only LLMs into text encoders. Muennighoff et al. (2025) introduce an additional training objective to preserve generative capabilities, but still require bidirectional attention. Alternatively, Echo embeddings proposed by Springer et al. (2025) avoid architecture modifications and allow a unified model for embedding and generation. After repetition, the output embedding of the final token is adopted as the representation of the input text, consistent with other auto-regressive methods (Li et al., 2025; Springer et al., 2025; Li et al., 2024a; Wang et al., 2024).

LLM-based retrieval. LLM-based embedding models offer a strong backbone for retrieval systems, facilitating more precise modeling of complex relationships between queries and documents. Repllama (Ma et al., 2024) fine-tuned LLaMA-2 to function as both a retriever and a reranker, showcasing the potential of large language models in retrieval pipelines. Llama2Vec (Li et al., 2024a) further improved performance by introducing two pretext tasks, achieving significant gains on the BEIR (Thakur et al., 2021) benchmark. These methods similarly adopt the embedding of the final token as the overall representation of queries and documents. However, they overlook the discrepancy between the [EOS] token's role in language model pre-training, where it functions merely as a sequence terminator, and its intended use as a semantic bridge in retrieval tasks, leaving the learning of more effective and query-document-aligned [EOS] embeddings an open challenge.

Auto-Reconstruction Methods. Prior works such as SimLM (Wang et al., 2023), LexMAE (Shen et al., 2022), RetroMAE (Xiao et al., 2022), and Condenser (Gao and Callan, 2021) have explored auto-reconstruction objectives to improve text embeddings for encoder-based models. To enrich the representations, these methods typically try to recover masked tokens or spans from intermediate encoder states. They rely on auxiliary decoders or reconstruction heads that are used during pre-training but discarded at inference, which often leads to complex multi-stage training pipelines.

3 Method

In this section, we present our two-stage training framework. The first stage is a novel training phase introduced in this work, which incorporates two bidirectional reconstruction tasks, detailed in Section 3.2. The second stage employs contrastive learning to further refine the representations, as described in Section 3.3. An overview of the entire pipeline is illustrated in Figure 1.

3.1 Preliminary

Language models (LMs) have been widely adopted as powerful embedding models in a variety of NLP tasks. A decoder-only language model $\mathcal{M}$ typically consists of an input embedding layer $\texttt{Embed}(\cdot)$, L stacked Transformer decoder blocks $\texttt{Dec}_\ell(\cdot)$ with self-attention modules, and a linear output head lm_head that projects the final hidden states to vocabulary logits for next-token prediction.

To obtain the embedding for an input sequence from LMs, we adopt a widely used approach that leverages the final hidden state. Specifically, given an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, we append the special end-of-sequence token [EOS] to form the full input. The hidden state at the position of [EOS] extracted from the final decoder layer is taken as the sequence embedding, formally defined as:

$$e_X = f(x_1, x_2, \ldots, x_n, \texttt{[EOS]})[-1], \quad (1)$$

where $f(\cdot)$ denotes the composition of the embedding process and the forward pass through all L stacked decoder layers, as computed below:

$$f(\cdot) = \mathtt{Dec}_{\ell}(\mathtt{Dec}_{\ell-1}(\ldots\mathtt{Dec}_{1}(\mathtt{Embed}(\cdot)))). \quad (2)$$

The objective of text retrieval is to find the top-k documents from a large-scale corpus that are most relevant to a given query. The semantic relevance between a query Q and a candidate document D is typically measured by the similarity between their embeddings, such as cosine similarity or the inner product, i.e., $<e_Q, e_D>$.

3.2 Stage I: Bidirectional Reconstruction

We introduce a novel training stage inserted between general LM pre-training and contrastive learning. During this stage, the model is supervised by two dual reconstruction objectives, namely EBQ2D (Embedding-Based Query-to-Document) and EBD2Q (Embedding-Based Document-to-Query), which guide the model to incorporate counterpart information into the output embedding. Accurate query-document pairs are used as the training corpus. The implementation details of the bidirectional reconstruction are illustrated in Algorithm 1.

EBQ2D. Given a natural language query, the EBQ2D objective aims to encourage the model to generate the relevant document conditioned on the query embedding. During training, the model first computes the embedding e_Q from the query token sequence $Q = \{q_1, \ldots, q_n\}$, as described in Equation 1. This embedding is then used as a prefix to condition the generation of document tokens $D = \{d_1, \ldots, d_m\}$ under the teacher forcing paradigm. The training objective minimizes the cross-entropy loss between the predicted and ground-truth tokens. This reconstruction of documents via query embedding requires the embedding to capture the explicit semantic meaning of the query while simultaneously integrating information indicative of the relevant document, which is essential for effectively retrieving relevant documents in subsequent tasks. Formally, the EBQ2D loss can be given by:

$$\mathcal{L}_{\text{Q2D}} = -\sum_{t=1}^{m} \log P_{\Theta}(d_t \mid e_Q, d_{<t}). \quad (3)$$

where Θ denotes the model parameters.

```
Algorithm 1 Bidirectional Reconstruction
Require: Paired data $(Q, D)$; model $\mathcal{M} = \mathtt{lm\_head} \circ \mathtt{Dec} \circ \mathtt{Embed}$
$e_Q \leftarrow f(Q)$; $e_D \leftarrow f(D)$
        ▷ Obtain embeddings as in Equation 1
$\mathbf{E}_Q \leftarrow \mathtt{Embed}(Q)$; $\mathbf{E}_D \leftarrow \mathtt{Embed}(D)$
$\hat{D} \leftarrow \mathtt{lm\_head}(\mathtt{Dec}([\![e_Q, \mathbf{E}_D]\!]))$
        ▷ Decode $D$ from $e_Q$ via teacher forcing
$\mathcal{L}_{\text{Q2D}} \leftarrow \text{CE}(\hat{D}, D)$
$\hat{Q} \leftarrow \mathtt{lm\_head}(\mathtt{Dec}([\![e_D, \mathbf{E}_Q]\!]))$
        ▷ Decode $Q$ from $e_D$ via teacher forcing
$\mathcal{L}_{\text{D2Q}} \leftarrow \text{CE}(\hat{Q}, Q)$
return $\mathcal{L}_{\text{StageI}} = \alpha\mathcal{L}_{\text{Q2D}} + (1-\alpha)\mathcal{L}_{\text{D2Q}}$
```

EBD2Q. Complementary to EBQ2D, the EBD2Q objective guides the model to recover the underlying user intent from the given document. The model first encodes the document and uses the representation of the `[EOS]` token, denoted as e_D, as the embedding that summarizes the document's content. Conditioned on e_D, the model generates the corresponding query auto-regressively, following a similar decoding process as in EBQ2D. This objective encourages the document embedding to capture high-level abstractions and latent intent signals necessary to reconstruct the query, enhancing the bidirectional alignment between queries and documents. Similarly, the EBD2Q loss is defined as:

$$\mathcal{L}_{\text{D2Q}} = -\sum_{t=1}^{n} \log P_{\Theta}(q_t \mid e_D, q_{<t}). \quad (4)$$

Training Stage I integrates the two tasks within a multi-task learning framework, where the overall objective is formulated as a weighted sum of $\mathcal{L}_{Q2D}$ and $\mathcal{L}_{D2Q}$:

$$\mathcal{L}_{\text{StageI}} = \alpha\mathcal{L}_{\text{Q2D}} + (1-\alpha)\mathcal{L}_{\text{D2Q}} \quad (5)$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the relative importance of the two objectives. As our experiments show that the performance is not sensitive to the choice of α, we fix it to 0.2, which yields slightly better results.

3.3 Stage II: Contrastive Learning

After bidirectional reconstruction training, the model is fine-tuned on downstream tasks through contrastive learning. In line with prior work (BehnamGhader et al., 2024; Springer et al., 2025), we use a replication of the public portion of the E5 dataset (Wang et al., 2024) as the training corpus for fine-tuning. The training process is guided by the widely used InfoNCE (Izacard et al., 2021) loss function $\mathcal{L}$:

$$\mathcal{L} = -\log \frac{\exp\left(\text{sim}\left(Q, D^{+}\right)\right)}{\exp\left(\text{sim}\left(Q, D^{+}\right)\right) + \sum_{j} \exp\left(\text{sim}\left(Q, D_{j}^{-}\right)\right)} \tag{6}$$

In this equation, D_j^- denotes the set of negatives, encompassing both in-batch and hard negatives. The matching score between a query Q and a document D is computed using a temperature-scaled cosine similarity function, defined as:

$$\text{sim}(Q, D) = \frac{1}{\tau}\cos(e_Q, e_D) \tag{7}$$

where τ is a temperature hyperparameter, which is fixed at 0.05 in our practice.

After our two-stage training, the embeddings generated by the model are enriched with semantic information and demonstrate improved alignment between queries and relevant documents. We refer to these enhanced representations as "Anchor Embeddings", which also denotes our proposed method.

4 Experiment

4.1 Basic Settings

Language Models. We apply our two-stage training framework to five decoder-only LLMs ranging from 1B to 8B parameters: Meta-LLaMA-3.2-1B-Instruct, Qwen2.5-1.5B-Instruct, Meta-LLaMA-3.2-3B-Instruct, Mistral-7B, and Meta-LLaMA-3.1-8B-Instruct.

Training Datasets and Setup. The public portion of the E5 dataset (Wang et al., 2024), curated by Springer et al. (2025), serves as the training corpus for both stages. It comprises roughly 1.5 million samples, with further details on its construction provided in A.1. We train the model with full-parameter tuning for 2000 steps in Stage I, and fine-tune it using LoRA for 1000 steps as done in LLM2Vec (BehnamGhader et al., 2024) in Stage II. We provide other hyper-parameters in Appendix A.2.

Benchmark. We evaluate our method on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), a collection of 56 datasets covering seven types of embedding tasks: classification, clustering, pairwise classification, reranking, retrieval, sentence similarity (STS), and summarization. A comprehensive description of MTEB is provided in Appendix B.1. Since MTEB is massive and requires multiple days to evaluate, we conduct ablations on a 15-task subset as adopted in LLM2Vec, with details provided in Appendix B.2.

Baselines. Since different models are often trained on diverse datasets, and many do not disclose the specific data used, for a fairer comparison and to more accurately assess the impact of our proposed training strategy, we conduct evaluations comparing against models trained solely on the publicly available data under zero-shot settings. The compared methods include earlier encoder-based models, such as Instructor-xl (Su et al., 2023) (1.5B) and BGE-large-en-v1.5 (Xiao et al., 2024) (335M). In addition, we compare against recent state-of-the-art approaches, including GritLM (Muennighoff et al., 2024), E5 (Wang et al., 2024), bge-en-icl (Li et al., 2025), and the fine-tuned Echo embedding (Springer et al., 2025), all of which are built upon Mistral-7B. For LLM2Vec (BehnamGhader et al., 2024), we compare against its `Bi + MNTP` variants built on S-LLaMA-1.3B, Mistral-7B, and Meta-LLaMA-3-8B, which achieve the best performance after supervised fine-tuning.

4.2 Main Results

Table 1 presents the performance of our method (marked as Anchor) compared to baselines trained solely with contrastive learning, without incorporating the proposed bidirectional reconstruction stage. All evaluations are conducted on the MTEB benchmark, with improvements over the baseline indicated as subscripts. Table 2 further compares our method against other state-of-the-art models on MTEB. Based on these tables, we analyze the results from the following three perspectives.

Firstly, the bidirectional reconstruction training consistently improves performance across different models and scales. For instance, compared to the baseline LLaMA-3.2-1B-Instruct model which is only fine-tuned with contrastive learning, Anchor$_{\text{LLaMA-3.2-1B-Instruct}}$ achieves an average score of 62.24%, with an absolute improvement of 1.25%. Especially, it shows substantial gains of 2.54% and 1.71% on the retrieval and re-ranking tasks, respectively. Similarly, Anchor$_{\text{Qwen2.5-1.5B-Instruct}}$ and Anchor$_{\text{Mistral-7B}}$ also demonstrate clear improvements, further validating the robustness of our method across different

Categories → # of datasets →	**Retr.** 15	**Rerank.** 4	**Clust.** 11	**PairClass.** 3	**Class.** 12	**STS** 10	**Summ.** 1	**Avg** 56
				LLaMA-3.2-1B-Instruct				
Baseline	50.06	54.94	44.38	82.71	72.17	81.27	29.94	60.99
Anchor (ours)	52.60$_{\mathbf{+2.54}}$	56.65$_{\mathbf{+1.71}}$	44.75$_{+0.37}$	85.48$_{+2.77}$	72.47$_{+0.30}$	82.10$_{+0.83}$	30.87$_{+0.93}$	62.24$_{\mathbf{+1.25}}$
				Qwen2.5-1.5B-Instruct				
Baseline	51.66	54.86	43.03	84.41	72.97	82.02	31.89	61.51
Anchor (ours)	53.62$_{\mathbf{+1.96}}$	57.63$_{\mathbf{+2.77}}$	43.19$_{+0.16}$	85.77$_{+1.36}$	74.51$_{+1.54}$	82.74$_{+0.72}$	31.61$_{-0.28}$	62.86$_{\mathbf{+1.35}}$
				LLaMA-3.2-3B-Instruct				
Baseline	51.72	56.13	43.40	86.11	74.67	82.73	30.98	62.33
Anchor (ours)	53.66$_{\mathbf{+1.94}}$	57.77$_{\mathbf{+1.64}}$	45.48$_{+2.08}$	86.53$_{+0.42}$	75.59$_{+0.92}$	82.48$_{-0.25}$	30.81$_{-0.17}$	63.55$_{\mathbf{+1.22}}$
				Mistral-7B				
Baseline	54.92	57.98	44.97	86.04	75.51	83.14	30.64	63.87
Anchor (ours)	56.87$_{\mathbf{+1.95}}$	60.56$_{\mathbf{+2.58}}$	45.73$_{+0.76}$	87.99$_{+1.95}$	75.95$_{+0.44}$	83.52$_{+0.38}$	30.28$_{-0.36}$	64.99$_{\mathbf{+1.12}}$
				LLaMA-3.1-8B-Instruct				
Baseline	55.36	58.92	46.64	86.80	74.80	83.10	29.67	64.06
Anchor (ours)	57.09$_{\mathbf{+1.73}}$	61.38$_{\mathbf{+2.46}}$	46.03$_{-0.61}$	88.92$_{+2.12}$	76.17$_{+1.37}$	83.76$_{+0.66}$	30.13$_{+0.46}$	65.30$_{\mathbf{+1.24}}$

Table 1: Performance on the MTEB benchmark. Baselines are trained only with regular contrastive learning (Stage II).

Categories → # of datasets →	**Retr.** 15	**Rerank.** 4	**Clust.** 11	**PairClass.** 3	**Class.** 12	**STS** 10	**Summ.** 1	**Avg** 56
Previous work w/ public data only								
Instructor-xl	49.26	57.29	44.74	86.62	73.12	83.06	**32.32**	61.79
BGE$_{\text{large-en-v1.5}}$	54.29	60.03	46.08	87.12	75.97	83.11	31.61	64.23
GritLM$_{\text{Mistral-7b-v1}}$ + public data	53.10	$\underline{61.30}$	**48.90**	86.90	$\underline{77.00}$	82.80	29.40	64.70
E5$_{\text{Mistral-7b-v1}}$ + public data	52.78	60.38	$\underline{47.78}$	$\underline{88.47}$	76.80	$\underline{83.77}$	$\underline{31.90}$	64.56
Echo$_{\text{Mistral-7b-v1}}$	55.52	58.14	46.32	87.34	**77.43**	82.56	30.73	64.68
bge-en-icl$_{\text{Mistral-7b-v1}}$ + E5 data (zero-shot)	**59.59**	56.85	42.61	87.87	75.47	83.30	29.52	64.67
LLM2Vec$_{\text{S-LLaMA-1.3B}}$	51.44	55.38	43.57	86.20	72.21	83.58	30.01	61.85
LLM2Vec$_{\text{Mistral-7B}}$	55.99	58.42	45.54	87.99	76.63	**84.09**	29.96	64.80
LLM2Vec$_{\text{Meta-LLaMA-3-8B}}$	56.63	59.68	46.45	87.80	75.92	83.58	30.94	$\underline{65.01}$
Anchor$_{\text{LLaMA-3.1-8B-Instruct}}$	$\underline{57.09}$	**61.38**	46.03	**88.92**	76.17	83.76	30.13	**65.30**

Table 2: Performance comparison on the MTEB benchmark with other advanced models. The best results for each subtask are highlighted in bold, and the second-best results are underlined.

model families. Notably, for larger models such as Mistral-7B, LLaMA-3.2-3B-Instruct, and LLaMA-3.1-8B-Instruct, our method continues to yield consistent gains, with average improvements of +1.12%, +1.22%, and +1.24%, respectively. This indicates that our bidirectional reconstruction training maintains its effectiveness as model size increases, which is essential to achieve superior performance with stronger models.

Secondly, our method establishes a new state-of-the-art performance on the MTEB. At the 1B scale, Anchor$_{\text{LLaMA-3.2-1B-Instruct}}$ and Anchor$_{\text{Qwen2.5-1.5B-Instruct}}$ achieve average scores of 62.24% and 62.86%, respectively, outperforming LLM2Vec$_{\text{S-LLaMA-1.3B}}$ at 61.85%. Likewise, Anchor$_{\text{Mistral-7B}}$ attains 64.99%, surpassing not only its contrastive baseline but also other competitive Mistral-7B-based models such as LLM2Vec (64.80%), GritLM (64.70%), E5 (64.56%), and Echo (64.68%), thereby confirming the effectiveness of our method under the same backbone architecture. At the larger scale, Anchor$_{\text{LLaMA-3.1-8B-Instruct}}$ reaches 65.30%, exceeding LLM2Vec with the same 8B scale at 65.01%, and establishing a new state-of-the-art performance. These results validate that our approach promotes the learning of richer semantic representations, yielding higher-quality embeddings and enhanced downstream task performance.

Thirdly, the bidirectional reconstruction tasks generally lead to performance improvements across most sub-tasks. The EBQ2D and EBD2Q objectives were specifically designed to capture implicit semantic relationships between queries and documents, primarily targeting retrieval and re-ranking tasks. Unexpectedly, we find that the reconstruction training maintains the model performance on other tasks and, in some cases, even delivers slight improvements. For instance, with LLaMA-3.2-1B-Instruct, our approach yields performance gains across all task categories, with the most notable improvements observed in retrieval and re-ranking as expected. This suggests that by learning to reconstruct the counterpart text from the [EOS] embedding, the model is encouraged to encode a more compact and semantically rich representation of the input, while better capturing relational information. For instance, for pair classification task, it becomes more effective at representing the relationship between sentence pairs, aiding in tasks such as assessing semantic similarity.

4.3 Ablation Study

Ablation on Reconstruction Tasks. We evaluate the effectiveness of the two reconstruction tasks proposed in Stage I, using the LLaMA-3.2-1B-Instruct model. The results are summarized in Table 3. The baseline refers to models trained exclusively with contrastive learning (i.e., without the bidirectional reconstruction stage), while "OnlyD2Q" and "OnlyQ2D" represent models trained with only one of the two reconstruction objectives during Stage I. Compared to the baseline, both the OnlyD2Q and OnlyQ2D variants yield consistent improvements across most tasks. In particular, the OnlyQ2D variant achieves higher scores. This indicates that the challenge of generating comprehensive documents from brief queries promotes the model to produce more expressive informative text embeddings. Moreover, our proposed method, which integrates both objectives, achieves the best overall performance with the highest average score of 62.24.

To investigate the impact of the hyperparameter α, we conduct experiments on the MTEB subset with α set to 0.2, 0.5, and 0.8. As shown in Table 4, the model exhibits relatively stable performance across these values, with our selected value $\alpha = 0.2$ yielding the best results.

Models → α →	**Baseline** -	**OnlyD2Q** 0	**OnlyQ2D** 1	**Anchor** 0.2
Retr.	50.06	51.54	52.05	**52.60**
Rerank.	54.94	56.30	56.62	**56.65**
Clust.	44.38	43.59	44.04	**44.75**
PairClass.	82.71	85.12	**85.50**	85.48
Class.	72.17	72.26	71.83	**72.47**
STS	81.27	80.70	81.04	**82.10**
Summ.	29.94	**30.91**	30.24	30.87
Avg	60.99	61.38	61.62	**62.24**

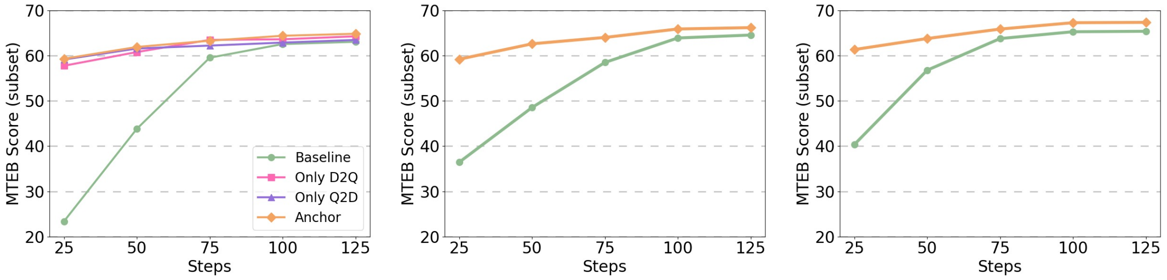
Table 3: Ablation study of training objectives in Stage I over the full MTEB benchmark.

α	**Average Score**
0.2	**65.19**
0.5	64.88
0.8	65.12

Table 4: Performance under different α values.

Early-stage During Fine-tuning. We save checkpoints every 25 steps and evaluate on the 15-task MTEB subset to assess early-stage fine-tuning performance. As shown in Figure 2, models trained with our bidirectional reconstruction tasks consistently demonstrate stronger performance in the early stages of fine-tuning in Stage II across all model sizes. The models trained after Stage I are nearly converged at the very beginning of fine-tuning, as further evidenced by the loss curves in Figure 4 provided in the Appendix. This suggests that our training with bidirectional reconstruction tasks is sufficiently powerful to endow the model with high-quality textual representations, reducing the reliance on subsequent contrastive learning.

Training stage. We conduct an ablation study to assess the contribution of each training stage, with the results plotted in Figure 3. The unsupervised baseline yields relatively low performance, suggesting that without task-specific supervision, the [EOS] output embeddings are insufficient for downstream tasks. Applying only Stage I with bidirectional reconstruction tasks results in a notable improvement, demonstrating that the model begins to learn semantic alignment through this training process. Incorporating Stage II leads to a further improvement, highlighting the indispensable role of contrastive learning in producing high-quality embeddings. Our two-stage framework significantly outperforms both single-stage variants:



(a) LLaMA-3.2-1B-Instruct (b) LLaMA-3.2-3B-Instruct (c) LLaMA-3.1-8B-Instruct

Figure 2: Results on the 15 task subset of MTEB during the first 125 training steps for LLaMA-3.2-1B-Instruct, LLaMA-3.2-3B-Instruct, and LLaMA-3.1-8B-Instruct.

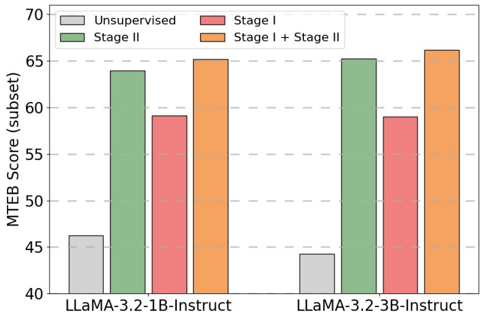


Figure 3: Impact of the two training stages on performance.

Training Configuration	Steps	Score
Contrastive only (Stage II)	1,000	63.95
Contrastive only (Stage II)	3,000	63.74
Full method (Stage I + II)	2,000 + 1,000	**65.19**

Table 5: Comparison of training strategies on the MTEB subset using LLaMA-3.2-1B-Instruct.

Stage I helps the model encode implicit semantic alignment between relevant texts, while Stage II further refines the representation space by bringing similar instances closer and pushing dissimilar ones apart.

Effectiveness beyond extra training. To rule out the possibility that the performance gains of our method stem merely from additional training steps, we evaluated a checkpoint trained solely on Stage II for 3,000 steps using the LLaMA-3.2-1B-Instruct model on an MTEB subset. The result was 63.74 as shown in Table 5, which is notably lower than the 65.19 achieved by our full method (Stage I: 2,000 steps + Stage II: 1,000 steps). Interestingly, the 3,000-step baseline even underperforms the 1,000-step version (63.95), which supports the rationale behind following LLM2Vec in reporting results after 1,000 steps of contrastive learning. These findings indicate that the observed improvement arises from the effectiveness of our bidirectional reconstruction training, rather than simply from longer training.

5 Conclusion

In this paper, we propose a two-stage training procedure for applying LLMs on text embedding tasks. Our method is designed to address the mismatch between the role of the [EOS] token in language model pre-training and downstream embedding tasks. We introduce two bidirectional reconstruction objectives, EBQ2D and EBD2Q, which treat the [EOS] output embedding as an anchor to aggregate semantic information from either queries or relevant documents to reconstruct their counterparts. This stage encourages the model to encode semantic alignment directly into the [EOS] representation. Our method achieves state-of-the-art performance on MTEB among models trained on the same publicly available datasets under zero-shot settings. We hope that our method offers valuable insight to advance the development and application of embedding models.

Limitations

While our method has effectively enhanced the performance of LLMs as text embedders, the current work can still be improved in the following ways. First, the query-document bidirectional reconstruction tasks in Stage I are primarily designed to benefit retrieval and re-ranking. For broader embedding tasks such as classification or clustering, more tailored objectives may yield further improvements. Secondly, although the first-stage training accelerates convergence in the second-stage fine-tuning and can reduce its computational cost, the overall two-stage training framework still introduces additional overhead. Future work should investigate how to improve training efficiency in order to support scaling to larger models. Thirdly, the current model is primarily developed for English-centric scenarios, and expanding its applicability to other languages remains an important future direction.

Ethical Considerations

Although our approach improves text embedding quality, it does not eliminate potential ethical risks associated with large language models. First, our approach may still inherit biases present in the training data, which can be amplified during the representation learning process, especially in domain-specific or underrepresented contexts. These biases can negatively influence downstream applications, such as search or recommendation systems. Additionally, like other LLM-based methods, our model may generate misleading or hallucinated outputs when used in generative scenarios, posing risks in high-stakes applications. We encourage responsible use and further evaluation of the model's behavior, particularly in sensitive domains.

Acknowledgments

This research work has been sponsored by National Key Research and Development Program of China (No. 2023ZD0121402), Ant Group Security and Risk Management Fund, and National Natural Science Foundation of China (NSFC) grant (No. 62576211).

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. 2017. Quora question pairs. https://kaggle.com/competitions/quora-question-pairs. Kaggle.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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.

Yasuhiro Fujiwara, Yasutoshi Ida, Atsutoshi Kumagai, Masahiro Nakano, Akisato Kimura, and Naonori Ueda. 2023. Efficient network representation learning via cluster similarity. *Data Science and Engineering*, 8(3):279–291.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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*, page 6894. Association for Computational Linguistics.

Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. 2023a. Jina embeddings: A novel set of high-performance sentence embedding models. *arXiv preprint arXiv:2307.11224*.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, and 1 others. 2023b. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, and 1 others. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics.

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*.

Mandar Joshi, Eunsol Choi, Daniel 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)*. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, and 1 others. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chaofan Li, Zheng Liu, 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.

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*.

Shiyu Li, Yang Tang, Shizhe Chen, and Xi Chen. 2024b. Conan-embedding: General text embedding with more and better negative samples. *arXiv preprint arXiv:2408.15710*.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

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, Hongjin SU, 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, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*.

Niklas Muennighoff, Nouamane Tazi, Loic 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. 2016. Ms marco: A human-generated machine reading comprehension dataset.

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.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2022. Lexmae: Lexicon-bottlenecked pretraining for large-scale retrieval. *arXiv preprint arXiv:2208.14754*.

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.

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

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)*. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 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, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 538–548.

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 1 others. 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.

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

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. 2023. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

A Training Details

A.1 Training Datasets

The public portion of the E5 dataset consists of ELI5 (sample ratio 0.1) (Fan et al., 2019), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), MIRACL (Zhang et al., 2023), MS-MARCO passage ranking (sample ratio 0.5) and document ranking (sample ratio 0.2) (Nguyen et al., 2016), NQ (Karpukhin et al., 2020), 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).

During Stage I, we use (query, positive) pairs as the training corpus. For fine-tuning in Stage II, we follow the setup of Wang et al. (2024), with dataset-specific instructions summarized in Table 6.

A.2 Training Setup

We adopt a two-stage training framework, where the model first undergoes full-parameter training with bidirectional reconstruction objectives, followed by parameter-efficient fine-tuning using LoRA. All experiments are performed with

Dataset	Instruction(s)
NLI	Given a premise, retrieve a hypothesis that is entailed by the premise Retrieve semantically similar text
DuReader	Given a Chinese search query, retrieve web passages that answer the question
ELI5	Provided a user question, retrieve the highest voted answers on Reddit ELI5 forum
FEVER	Given a claim, retrieve documents that support or refute the claim
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
MIRACL	Given a question, retrieve Wikipedia passages that answer the question
MrTyDi	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
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
SQuAD	Retrieve Wikipedia passages that answer the question
T2Ranking	Given a Chinese search query, retrieve web passages that answer the question
TriviaQA	Retrieve Wikipedia passages that answer the question

Table 6: Instructions used for each of the E5 datasets during fine-tuning.

a maximum input length of 512 tokens and FlashAttention-2 enabled.

Stage I. In training stage I, models are trained for 2000 steps on the E5 dataset using only (query, positive) pairs. The learning rate is set to $4e-5$, with a total batch size of 512 achieved by gradient accumulation. We apply linear warm-up over the first 300 steps and use end-of-sequence ([EOS]) token pooling to obtain sentence embeddings. Gradient checkpointing is enabled to reduce memory usage.

Stage II. Fine-tuning in Stage II is performed for 1000 steps with LoRA (rank 16) using a learning rate of $2e-4$ and a total batch size of 512. The same EOS pooling strategy and gradient checkpointing settings are the same as Stage I. Instructions used for each of the E5 dataset during this stage are summarized in Table 6.

A.3 Fine-tuning Loss

Figure 4 presents the Gaussian-smoothed training loss curves during Stage II fine-tuning of LLaMA-3.2-3B-Instruct. Models that have gone through Stage I (bidirectional reconstruction training) start with significantly lower losses, dropping from over 10 to below 1, compared to models trained directly in Stage II. This demonstrates the effectiveness of our bidirectional reconstruction training. In addition, models trained with Stage I converge faster and more smoothly during Stage II.

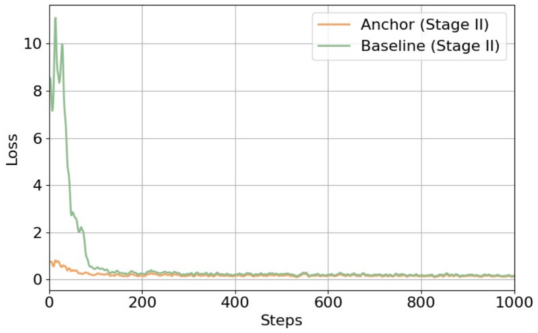


Figure 4: Comparison of training loss during fine-tuning: Baseline vs. Training Stage I Initialization (Anchor).

Model Size	Stage I	Stage II
1B	$\approx 4hrs$	$\approx 2hrs$
3B	$\approx 12hrs$	$\approx 5hrs$
8B	$\approx 45hrs$	$\approx 13hrs$

Table 7: Training time (in hours) for each stage.

A.4 Training Efficiency of Each Stage

To provide a clearer view of the computational cost involved in our two-stage training pipeline, we report the training time required for each stage across different model sizes. Specifically, we evaluate the time taken by Stage I (bidirectional reconstruction pretraining) and Stage II (contrastive fine-tuning) on 8×80GB NVIDIA A100 GPUs, with the results summarized in Table 7.

B Massive Text Embeddings Benchmark (MTEB)

B.1 Task Overview

The MTEB benchmark consists of a broad range of embedding tasks, including classification, clustering, pairwise classification, re-ranking, retrieval, sentence similarity (STS), and summarization, aiming to provide a comprehensive and robust evaluation of embedding quality. For evaluation, we follow the instruction templates from Wang et al. (2024), as shown in Table 13.

B.2 MTEB Subset

To speed up the evaluation, we follow the approach of LLM2vec and adopt the same representative subset of 15 tasks from MTEB for our analyses, as shown in Table 8. This subset was carefully selected to maintain a similar proportional distribution across categories compared to the full MTEB benchmark, ensuring that ablation studies and analyses are not biased toward any specific category or task.

Category	**Dataset**
Retrieval (3)	SciFact ArguAna NFCorpus
Reranking (2)	StackOverflowDupQ. SciDocsRR
Clustering (3)	BiorxivClusteringS2S MedrxivClusteringS2S TwentyNewsgroupsClus.
Pair Classification (1)	SprintDuplicateQ.
Classification (3)	Banking77Classification EmotionClassification MassiveIntentClassification
STS (3)	STS17 SICK-R STSBenchmark
SummEval (0)	-
Overall	15 datasets

Table 8: Subset of MTEB tasks for ablation studies.

C LLM-Based Baselines

E5 (Wang et al., 2024) trains text embedding models by fine-tuning open-source decoder-only LLMs using synthetic data generated by proprietary large language models. The synthetic corpus covers a wide range of embedding tasks across 93 languages and is created entirely without human annotation. During training, a standard contrastive loss is applied to learn effective representations from these synthetic text pairs. We only compare E5 results trained on the publicly available portion of the dataset.

GritLM (Muennighoff et al., 2024) unifies representation and generation training by finetuning a decoder-only LLM on instruction-formatted data for both tasks. It optimizes a contrastive loss for embeddings:

$$\mathcal{L}_{\text{Rep}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp\left(\tau\cdot\sigma(f_\theta(q^{(i)}), f_\theta(d^{(i)}))\right)}{\sum_{j=1}^{M}\exp\left(\tau\cdot\sigma(f_\theta(q^{(i)}), f_\theta(d^{(j)}))\right)}$$

and a standard language modeling loss for generation:

$$\mathcal{L}_{\text{Gen}} = -\frac{1}{N}\sum_{i=1}^{N}\log P(f_{\theta,\eta}(x^{(i)}) \mid f_{\theta,\eta}(x^{(<i)})).$$

The final objective combines both:

$$\mathcal{L}_{\text{GRIT}} = \lambda_{\text{Rep}}\mathcal{L}_{\text{Rep}} + \lambda_{\text{Gen}}\mathcal{L}_{\text{Gen}}.$$

This approach enables parameter-efficient training of strong text encoders and generators without modifying model architecture. Similarly, we only compare results trained on publicly available data.

Echo Embeddings (Springer et al., 2025) derives high-quality text embeddings from autoregressive language models without modifying their architecture. The key idea is to repeat the input sequence and extract embeddings from the repeated tokens, which have access to the full context of the original input. This simple repetition strategy enables autoregressive models to approximate bidirectional behavior and significantly improves embedding quality in zero-shot settings.

LLM2Vec (BehnamGhader et al., 2024) converts decoder-only language models into effective text encoders by enabling bidirectional attention, training with masked next token prediction, and applying contrastive learning.

bge-en-icl (Li et al., 2025) enhances the text representation ability of decoder-only language models by leveraging in-context learning during training. Instead of relying on task-specific instructions

Steps	**25**	**50**	**75**	**100**	**125**
LLaMA-3.2-1B-Instruct					
Baseline	23.38	43.83	59.55	62.56	63.07
OnlyD2Q	57.77	60.80	63.49	63.62	64.28
OnlyQ2D	59.15	61.58	62.23	62.89	63.48
Anchor	**59.31**	**61.92**	**63.31**	**64.40**	**64.84**
LLaMA-3.2-3B-Instruct					
Baseline	36.43	48.54	58.50	63.91	64.56
Anchor	**59.21**	**62.62**	**64.03**	**65.90**	**66.20**
LLaMA-3.1-8B-Instruct					
Baseline	40.35	56.79	63.78	65.29	65.39
Anchor	**61.35**	**63.79**	**65.88**	**67.29**	**67.36**

Table 9: Early-stage performance on the MTEB subset during Stage II fine-tuning.

or architectural modifications, it samples a variable number of input examples to simulate in-context scenarios. This strategy equips the model with the ability to generalize across tasks while preserving zero-shot performance. For fair comparison, we only compare results from models trained on the same data and evaluated in zero-shot settings.

D Experimentals

D.1 Detailed Main Results

We report the detailed performance of baselines and Anchor Embedding models built on LLaMA-3.2-1B-Instruct, Qwen2.5-1.5B-Instruct, LLaMA-3.2-3B-Instruct, and LLaMA-3.1-8B-Instruct across the full MTEB benchmark in Table 14 and Table 15. Here, baselines refer to models trained only with Stage II.

D.2 Early-stage Fine-tuning Results

In Table 9, we report the detailed evaluation scores on the 15-task MTEB subset at checkpoints saved every 25 steps during Stage II fine-tuning. Anchor Embedding consistently outperforms the baselines across all model sizes and converges faster.

D.3 Results of Training Stage Ablation

To support the analysis in Figure 3, we present the exact performance scores of different training stage combinations on the 15-task MTEB subset. As shown in Table 10, the unsupervised baseline yields the lowest scores, while adding either Stage I (bidirectional reconstruction) or Stage II (contrastive fine-tuning) brings significant gains. The best results are achieved when combining both stages, confirming the effectiveness of our method.

Training Setting	**1B**	**3B**
Unsupervised Baseline	46.23	44.29
Only Stage I (Bi-Reconstruction)	59.11	59.01
Only Stage II (Contrastive)	63.95	65.24
Stage I + Stage II	**65.19**	**66.16**

Table 10: Impact of training stages on performance evaluated on MTEB subset.

E Comparison with LLaMA2Vec

To further clarify our method, we provide an extended discussion of the differences between our Anchor and previous related work LLaMA2Vec (Li et al., 2024a).

E.1 Methodology

Although both Anchor and LLaMA2Vec adopt reconstruction-based objectives to improve the [EOS] embedding, their formulations and training paradigms differ substantially.

Reconstruction mechanism. LLaMA2Vec treats reconstruction as a multi-class classification problem: the [EOS] embedding is directly projected through the LLM's output head into the vocabulary space to perform token classification. Specifically, the objective function of this problem is derived as:

$$\min -\frac{1}{|T|}\sum_{t\in T}\log\frac{\exp(e^T W_t)}{\sum_{v\in V}\exp(e^T W_v)}$$

where $W \in R^{d\times|V|}$ is the projection head of LLM; V indicates the vocabulary space; T stands for the collection of tokens of the target context (input text itself for e_t^{α}, the next sentence for e_t^{β}).

In contrast, Anchor adopts a bidirectional generative strategy. We reuse the [EOS] embedding from either the query or document as input and apply teacher-forcing language modeling to generate its counterpart sequence (document or query). The training loss is standard cross-entropy over the generated sequence.

Supervision signal. LLaMA2Vec operates in a fully unsupervised setting, relying only on raw text.

Model	Steps (Stage I)	BEIR (N@10)
LLaMA2Vec	10,000	56.40
Anchor	2,000	**58.07**

Table 11: Comparison of Anchor and LLaMA2Vec on the BEIR benchmark.

Anchor leverages task-relevant supervision from query-document pairs. While this introduces a data dependency, it also brings stronger semantic alignment: the [EOS] embedding acts as a semantic anchor that captures meaningful relationships between queries and documents, enhancing effectiveness for retrieval and re-ranking tasks.

E.2 Experiment

Training Efficiency. LLaMA2Vec requires 10,000 training steps in its unsupervised adaptation stage (batch size 256). In contrast, Anchor achieves better results with only 2,000 steps in Stage I (batch size 512). Importantly, Stage I can be trained on the same data used for fine-tuning, without requiring additional annotation.

Effectiveness. We report the performance on the BEIR benchmark using NDCG@10 in Table 11. Anchor outperforms LLaMA2Vec while requiring fewer training steps in Stage I, demonstrating both higher efficiency and stronger retrieval effectiveness.

F Related Works

A number of general-purpose embedding models from industry, such as Alibaba's GTE (Li et al., 2023), NVIDIA's NV-Embed (Lee et al., 2024), Tencent's Conan-Embedding (Li et al., 2024b), Google's Gemini Embedding (Lee et al., 2025), and the Jina Embeddings series (Günther et al., 2023a,b), have shown strong performance across retrieval and semantic tasks, and are widely used in real-world applications. However, these models are often built with large-scale proprietary data and engineering pipelines that make fair academic comparison difficult. Therefore, we do not directly compare with them, and instead include a brief technical overview.

The method proposed in Li et al. (2023) introduces a lightweight 110M-parameter Transformer encoder enhanced with rotary positional encodings and gated linear units. It follows a two-stage contrastive training process—unsupervised pretraining on large-scale web corpora followed by supervised fine-tuning on relevance datasets using InfoNCE loss. Despite its compact size, this approach outperforms many larger models on retrieval and classification benchmarks.

NV-Embed (Lee et al., 2024) builds on a 7B decoder-only LLM architecture augmented with a latent-attention pooling layer. During contrastive training, the model discards causal masking, and a two-stage instruction tuning process with curated hard negatives further enhances the learned representations. This yields state-of-the-art results across tasks including semantic retrieval, semantic similarity, reranking, and dense passage retrieval.

Conan-Embedding (Li et al., 2024b) uses a 1.4B-parameter encoder trained with dynamic hard negative mining and a cross-GPU balancing loss for scalable negative sampling. The inclusion of LLM-generated prompt–response pairs as weak supervision allows the model to top the Chinese MTEB leaderboard and perform strongly in multilingual scenarios.

Gemini Embedding (Lee et al., 2025) fine-tunes a multilingual and multimodal LLM to produce 3,000-dimensional embeddings via contrastive learning on high-quality filtered datasets. To better support low-resource languages, it incorporates synthetic data. This design enables strong performance on cross-lingual and cross-modal retrieval benchmarks.

The Jina Embeddings framework (Günther et al., 2023a) employs T5-based encoder-only models, ranging from 35M to 6B parameters. These models are first trained on pairwise contrastive objectives using hundreds of millions of filtered sentence pairs, and then further refined via triplet-margin fine-tuning, with curated hard negatives including negation examples. The resulting embeddings outperform or match much larger models.

Jina Embeddings 2 (Günther et al., 2023b) adapts a BERT-style encoder with ALiBi and gated linear units to support input lengths up to 8,192 tokens. Its three-stage training—long-sequence masked language modeling, contrastive fine-tuning, and hard-negative refinement—produces embeddings competitive with OpenAI's ada-002 and establishes new benchmarks for long-document understanding.

Usage	Dataset	License / URL
Training	Public Portion of E5 (Wang et al., 2024), curated by Springer et al. (2025)	Apache License 2.0 https://github.com/jakespringer/echo-embeddings#training https://github.com/jakespringer/echo-embeddings/blob/master/LICENSE
Evaluation	MTEB Benchmark (Muennighoff et al., 2023)	Apache License 2.0 https://github.com/embeddings-benchmark/mteb https://github.com/embeddings-benchmark/mteb/blob/main/LICENSE

Table 12: License information for datasets used in this work.

G URLs and Licenses

Table 12 summarizes the license information for the datasets used. All datasets are employed strictly for research purposes and in compliance with their respective licenses and intended usage guidelines.

Task Name	Instruction
AmazonCounterfactualClassif.	Classify a given Amazon customer review text as either counterfactual or non-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 utterance as query, find the user scenarios
MTOPDomainClassification	Classify the internet 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
BUCC/Tatoeba	Retrieve parallel sentences
SummEval	Given a news summary, retrieve other semantically similar summaries

Table 13: Instructions used for MTEB evaluation. “STS*” denotes the set of all STS tasks.

Task	LLaMA-3.2-1B-Instruct		Qwen2.5-1.5B-Instruct	
	Baseline	Anchor	Baseline	Anchor
AmazonCounterfactualClassification	73.70	75.93	73.25	71.81
AmazonPolarityClassification	87.11	85.81	91.51	94.32
AmazonReviewsClassification	44.96	43.11	45.78	48.50
ArguAna	54.00	55.28	53.75	55.71
ArxivClusteringP2P	47.09	48.05	46.00	47.93
ArxivClusteringS2S	42.07	41.65	40.78	39.99
AskUbuntuDupQuestions	60.42	61.73	58.22	63.31
BIOSSES	83.91	85.87	84.67	86.14
Banking77Classification	85.63	85.65	81.73	84.62
BiorxivClusteringP2P	39.29	37.36	32.61	35.06
BiorxivClusteringS2S	35.52	33.78	32.94	31.74
CQADupstackRetrieval	41.04	42.78	44.79	45.28
ClimateFEVER	33.30	34.27	34.10	32.82
DBPedia	45.38	43.29	42.07	43.96
EmotionClassification	49.46	50.41	50.32	48.97
FEVER	88.48	88.48	87.85	86.61
FiQA2018	37.19	41.42	41.64	44.25
HotpotQA	59.24	69.48	64.69	67.96
ImdbClassification	72.90	76.61	76.06	88.31
MSMARCO	38.54	39.80	38.17	39.87
MTOPDomainClassification	94.02	94.51	91.55	94.52
MTOPIntentClassification	77.01	80.18	80.88	81.61
MassiveIntentClassification	76.21	74.97	77.96	76.05
MassiveScenarioClassification	79.31	79.00	79.94	77.55
MedrxivClusteringP2P	32.51	33.09	34.60	30.21
MedrxivClusteringS2S	31.06	29.95	32.46	29.32
MindSmallReranking	32.01	31.61	30.11	32.26
NFCorpus	35.98	35.87	37.29	37.15
NQ	54.62	57.66	56.52	60.31
QuoraRetrieval	88.22	88.83	88.46	89.17
RedditClustering	54.23	55.89	54.07	52.27
RedditClusteringP2P	60.88	61.98	60.85	58.10
SCIDOCS	18.07	20.10	20.53	20.40
SICK-R	80.70	81.64	81.76	82.10
STS12	71.32	73.18	76.10	76.65
STS13	85.02	84.36	85.72	85.12
STS14	79.80	79.80	80.29	80.94
STS15	84.98	85.48	87.36	87.81
STS16	84.49	85.68	84.24	85.27
STS17	91.28	90.67	89.80	90.88
STS22	65.15	67.27	64.57	66.12
STSBenchmark	86.08	87.01	85.67	86.38
SciDocsRR	77.85	81.84	80.93	83.43
SciFact	68.91	74.05	71.40	74.48
SprintDuplicateQuestions	88.18	95.31	94.30	94.94
StackExchangeClustering	65.70	66.08	63.21	67.80
StackExchangeClusteringP2P	30.86	34.42	30.44	35.23
StackOverflowDupQuestions	49.49	51.40	50.18	51.53
SummEval	29.94	30.87	31.89	31.61
TRECCOVID	70.86	78.21	74.27	83.36
Touche2020	17.07	19.44	19.35	23.00
ToxicConversationsClassification	64.40	62.18	64.83	65.90
TweetSentimentExtractionClassification	61.34	61.24	61.79	61.99
TwentyNewsgroupsClustering	48.97	50.04	45.40	47.43
TwitterSemEval2015	73.24	74.54	72.07	75.91
TwitterURLCorpus	86.70	86.58	82.55	86.46
Average	60.99	62.24	61.51	62.86

Table 14: Results of Anchor Emebeddings and baselines on MTEB.

Task	LLaMA-3.2-3B-Instruct		LLaMA-3.1-8B-Instruct	
	Baseline	Anchor	Baseline	Anchor
AmazonCounterfactualClassification	80.69	83.19	81.66	82.30
AmazonPolarityClassification	87.65	88.19	88.84	92.26
AmazonReviewsClassification	46.78	47.97	46.86	47.96
ArguAna	54.28	56.12	56.71	57.96
ArxivClusteringP2P	44.71	46.75	46.10	47.99
ArxivClusteringS2S	39.57	43.63	44.11	44.77
AskUbuntuDupQuestions	60.63	62.63	64.18	66.87
BIOSSES	85.79	86.22	86.98	85.00
Banking77Classification	86.99	86.74	87.18	87.43
BiorxivClusteringP2P	34.59	37.18	39.01	38.62
BiorxivClusteringS2S	33.72	34.95	36.19	37.29
CQADupstackRetrieval	41.38	44.20	48.37	49.22
ClimateFEVER	34.09	33.86	35.17	37.79
DBPedia	41.54	44.80	49.01	51.88
EmotionClassification	49.64	50.91	50.64	51.78
FEVER	88.55	90.82	89.12	91.68
FiQA2018	40.60	47.51	47.84	48.40
HotpotQA	68.51	65.37	75.22	77.12
ImdbClassification	80.11	81.31	82.69	84.22
MSMARCO	39.38	41.08	41.21	43.29
MTOPDomainClassification	95.02	95.81	92.15	94.57
MTOPIntentClassification	82.01	83.48	80.25	83.47
MassiveIntentClassification	77.70	78.36	79.06	78.72
MassiveScenarioClassification	79.89	80.89	81.12	80.99
MedrxivClusteringP2P	29.17	31.33	33.86	31.90
MedrxivClusteringS2S	27.56	29.75	31.72	31.14
MindSmallReranking	29.58	32.31	33.53	36.01
NFCorpus	38.47	36.62	39.82	39.82
NQ	57.80	62.30	62.16	64.16
QuoraRetrieval	88.53	89.18	89.80	88.85
RedditClustering	56.86	59.45	59.89	60.02
RedditClusteringP2P	62.35	64.07	61.85	63.86
SCIDOCS	18.18	17.78	21.97	23.12
SICK-R	82.37	82.57	81.22	83.38
STS12	76.50	76.81	76.84	76.98
STS13	85.53	83.93	83.42	86.81
STS14	81.56	81.18	81.63	83.27
STS15	87.90	85.84	87.85	88.33
STS16	85.86	84.63	86.31	86.72
STS17	89.99	91.36	91.82	91.34
STS22	65.29	65.57	67.15	67.25
STSBenchmark	86.49	86.69	87.78	88.55
SciDocsRR	82.98	83.17	84.87	86.93
SciFact	74.09	75.47	76.97	79.12
SprintDuplicateQuestions	96.29	95.69	94.10	96.51
StackExchangeClustering	69.87	69.41	68.04	68.35
StackExchangeClusteringP2P	32.38	32.72	31.26	31.22
StackOverflowDupQuestions	51.33	52.97	53.23	55.72
SummEval	30.98	30.98	29.67	30.13
TRECCOVID	71.45	80.46	79.30	81.02
Touche2020	18.89	19.29	17.96	22.85
ToxicConversationsClassification	67.36	68.07	64.96	68.03
TweetSentimentExtractionClassification	62.16	62.17	62.27	62.32
TwentyNewsgroupsClustering	46.64	51.07	50.55	51.18
TwitterSemEval2015	75.85	77.37	79.47	80.28
TwitterURLCorpus	86.19	86.52	86.83	89.98
Average	62.33	63.55	64.06	65.30

Table 15: Results of Anchor Emebeddings and baselines on MTEB.