Correcting Mean Bias in Text Embeddings: A Refined Renormalization with Training-Free Improvements on MMTEB

Xingyu Ren [*1] **Youran Sun** [*2] **Haoyu Liang** [3]

Abstract

We find that current text embedding models produce outputs with a consistent bias, i.e., each embedding vector e can be decomposed as $\tilde{e} + \mu$, where μ is almost identical across all sentences. We propose a plug-and-play, training-free and lightweight solution called *renormalization*. Through extensive experiments, we show that renormalization consistently and statistically significantly improves the performance of existing models on the Massive Multilingual Text Embedding Benchmark (MMTEB). In particular, across 38 models, renormalization improves performance by **9.7 sigma** on retrieval tasks, **3.1 sigma** on classification tasks, and **0.8 sigma** on other types of tasks. Renormalization has two variants: directly subtracting μ from e, or subtracting the projection of e onto μ. We theoretically predict that the latter performs better, and our experiments confirm this prediction.

1. Introduction

A *text embedding model* is a neural network that maps a piece of text t to a dense vector $e \in \mathbb{R}^d$ so that semantic similarity between texts is reflected by geometric proximity of their embeddings. Text embedding models are widely used in tasks such as classification, clustering, retrieval, and reranking. Previous studies (Liang et al., 2025; Li et al., 2024) have found that text embedding models generally exhibit a systematic bias. Specifically, when a text embedding model $\mathcal{E}$ is applied to a large set of texts $\{t_i\}$ with diverse content and languages, and we compute the mean embedding

$$\mu = \mathbb{E}_i\big[\mathcal{E}(t_i)\big], \tag{1}$$

this mean vector is significantly nonzero. Or, in other words, each embedding can be decomposed as

$$e = \tilde{e} + \mu, \tag{2}$$

where μ is almost identical across sentences, and $\tilde{e}$ is the true signal. Prior work (Liang et al., 2025) has used this property to construct attacks and hypothesized that the output bias degrades downstream task performance.

To address this issue, we propose a plug-and-play, training-free solution called *renormalization*. The idea is simple. Since μ is an irrelevant constant vector, we should remove it from the outputs of a text embedding model $\mathcal{E}$. Doing so has two desirable effects: (1) the model outputs become more uniformly distributed in the embedding space $\mathbb{R}^d$; (2) the noise in the μ direction is removed, so the model performance should improve. Concretely, we consider two removal methods.

R1: subtract μ directly and then normalize to unit length[1].

$$\tilde{e}_1 = e - \mu, \qquad e_1' = \frac{\tilde{e}_1}{\|\tilde{e}_1\|}. \tag{3}$$

R2: remove the component of e along μ and then normalize.

$$\tilde{e}_2 = e - (e \cdot \hat{\mu})\hat{\mu}, \qquad e_2' = \frac{\tilde{e}_2}{\|\tilde{e}_2\|}. \tag{4}$$

From a noise propagation perspective, we predict that R2 will perform better (see Sec. 2), and we verify this in our experiments (see Sec. 3).

The main contributions of this paper are as follows:

1. Formalizing and implementing two renormalization methods for sentence embeddings.

[*]Equal contribution [1]Dept. of Physics, The Chinese University of Hong Kong, Hong Kong, China [2]Dept. of Mathematical Sciences, Tsinghua University, Beijing, China [3]Dept. of Computer Science and Technology, Tsinghua University, Beijing, China. Correspondence to: Haoyu Liang <lianghy18@tsinghua.org.cn>.

[1]For most text embedding models (for example, all the models tested in this paper), the embedding is normalized to an unit vector. Therefore, we will assume this property throughout this paper. For those non-unit text embedding models, similar renormalization method can be applied.

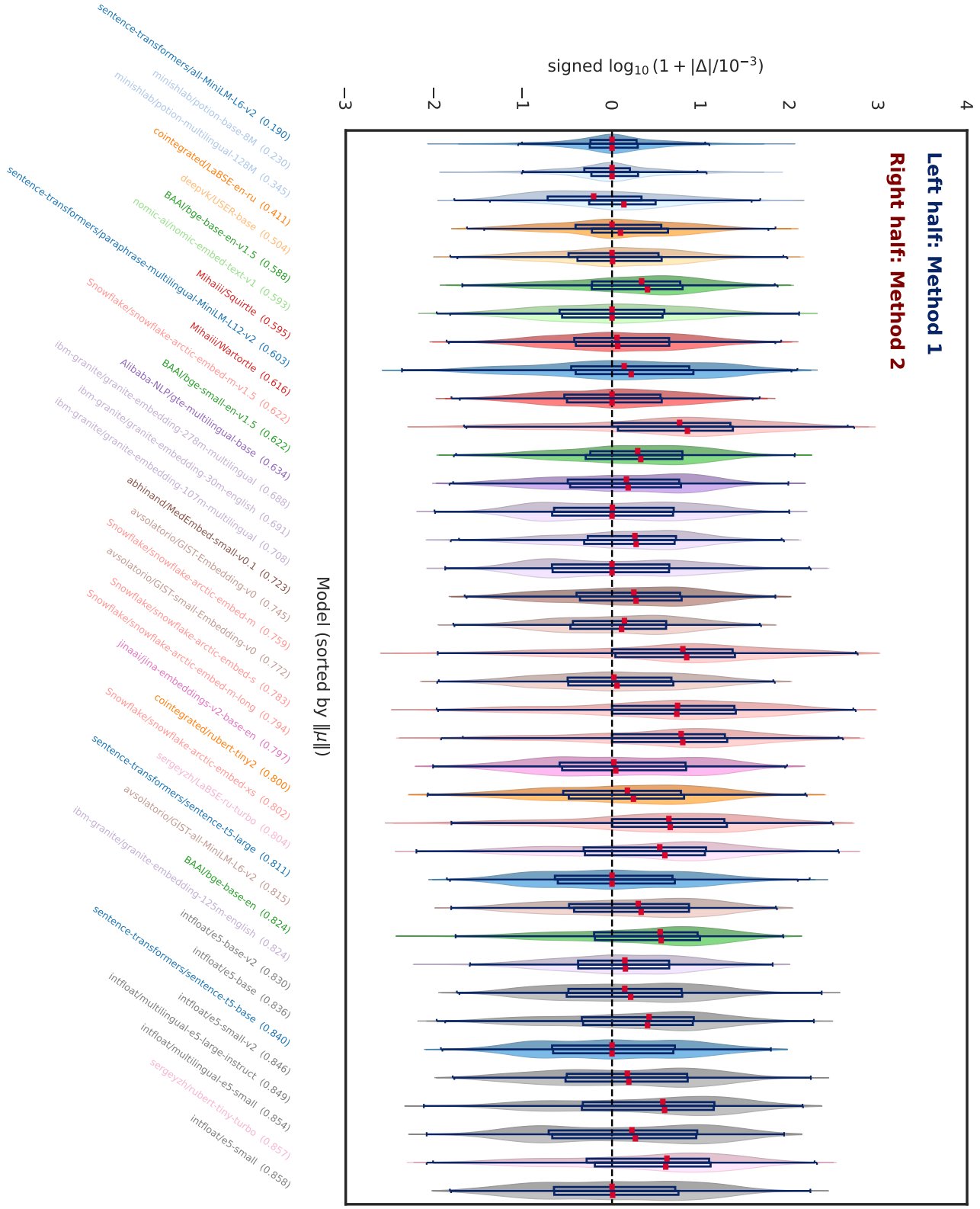

Figure 1. Comparison of two renormalization methods across models, sorted by mean vector norm (labeled after the model name). The violin shows the distribution of task score difference Δ before and after the renormalization. The median (red bar), quaters and upper and lower adjacent values are marked in the violin. The y-axis is properly scaled by logarithm to better present the distribution. The models in the same company use the same color, while in different companies use different colors.

2. Evaluating multiple public models on MMTEB to verify the effectiveness of the renormalization methods.
3. Linking gains of the renormalization method to the magnitude of the mean embedding vector, supporting the anisotropy explanation.

2. Method

Notation. Let $\mathcal{E}$ denote a text embedding model. For a piece of text t, the model produces an embedding vector $e \in \mathbb{R}^d$, i.e. $\mathcal{E}(t) = e$. Given a large collection of texts $\{t_i\}_{i=1}^N$, we denote the mean embedding (bias) by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathcal{E}(t_i), \tag{5}$$

or, when the underlying distribution is clear, $\mu = \mathbb{E}_t[\mathcal{E}(t)]$. The normalized mean is denoted by

$$\hat{\mu} = \frac{\mu}{\|\mu\|}. \tag{6}$$

Comparison of two renormalization methods. We previously introduced two variants of renormalization, **R1** and **R2**. The difference is whether we (R1) subtract μ directly from e or (R2) remove the projection of e onto $\hat{\mu}$, i.e. $(e\cdot\hat{\mu})\hat{\mu}$. From an error propagation perspective, we expect **R2** to be superior. The argument is as follows.

We estimate μ by averaging the embeddings $\mathcal{E}(t)$ over a large collection of texts; this estimator inevitably contains an estimation error ϵ. Assume the estimator of μ is $\mu + \epsilon$, where the estimation error ϵ decomposes into components parallel and orthogonal to μ

$$\epsilon = \epsilon_\parallel + \epsilon_\perp, \quad \epsilon_\parallel = (\epsilon \cdot \hat{\mu})\hat{\mu}, \quad \epsilon_\perp \cdot \mu = 0. \tag{7}$$

Denote the true signal by $\tilde{e} = e - \mu$. We further assume $\tilde{e}$ is approximately orthogonal to μ and to typical random directions in the orthogonal subspace.

For **R1** (subtract the estimated mean)

$$\tilde{e}_1 = \tilde{e} - \epsilon_\parallel - \epsilon_\perp. \tag{8}$$

For **R2** (remove the projection onto the estimated mean)

$$\tilde{e}_2 = \tilde{e} + \mu - \frac{(\tilde{e}+\mu)\cdot(\mu+\epsilon)}{\|\mu+\epsilon\|^2}(\mu+\epsilon). \tag{9}$$

Under the approximations $\tilde{e}\cdot\mu \approx 0$ and $\tilde{e}\cdot\epsilon \approx 0$ (high-dimensional near-orthogonality), the numerator and denominator can be simplified:

$$\begin{aligned}(\tilde{e}+\mu)\cdot(\mu+\epsilon) &\approx \|\mu\|^2 + \mu\cdot\epsilon = \|\mu\|^2 + \|\mu\|\|\epsilon_\parallel\|,\\ \|\mu+\epsilon\|^2 &= (\|\mu\| + \|\epsilon_\parallel\|)^2 + \|\epsilon_\perp\|^2.\end{aligned} \tag{10}$$

For $\|\epsilon\| \ll \|\mu\|$ a first-order expansion gives

$$\begin{aligned}\tilde{e}_2 &\approx \tilde{e} + \mu - \frac{\|\mu\|^2 + \|\mu\|\|\epsilon_\parallel\|}{(\|\mu\| + \|\epsilon_\parallel\|)^2 + \|\epsilon_\perp\|^2}(\mu + \epsilon_\parallel + \epsilon_\perp)\\ &\approx \tilde{e} + \mu - \left(1 - \frac{\|\epsilon_\parallel\|}{\|\mu\|}\right)(\mu + \epsilon_\parallel + \epsilon_\perp)\\ &= \tilde{e} - \epsilon_\perp.\end{aligned} \tag{11}$$

Thus, up to higher-order terms, **R2** cancels the parallel component $\epsilon_\parallel$ of the estimation error and only leaves the orthogonal component $\epsilon_\perp$, while **R1** retains both components. This explains why **R2** is expected to perform better.

3. Experiments

3.1. Experimental Setup

Computation of μ. For each model, we precompute a corpus-level mean vector μ using a large, disjoint Wikipedia snapshot (20220301.en). The corpus is sentence-segmented and filtered to include only sentences with lengths between 64 and 512 characters. A total of 100,000 sentences are randomly sampled to estimate μ. To prevent data leakage, we ensure that the evaluation data are not used in this computation.

Benchmarks. We evaluate on MMTEB (Enevoldsen et al., 2025), the multilingual extension of MTEB (Muennighoff et al., 2023), covering more than 500 quality-controlled tasks in 250+ languages. We follow the official splits and metrics.

Task filtering. Some MTEB tasks are excluded due to modality mismatch (e.g., image or multi-modal tasks for text-only encoders), prior hard failures, or repeated timeouts beyond a fixed per-task wall clock budget (default 1 hour, sometimes extended to 3 hours). The full enumerated list and rationale are provided in Appendix A, ensuring transparency in aggregate score comparisons.

Models. We evaluate a representative set of publicly available embedding models from the MTEB leaderboard, spanning different parameter scales and training paradigms. See Appendix B for the models we used.

3.2. Main Result

As shown in Table 1, the renormalization methods (both R1 and R2) achieve significant improvements across models from multiple companies. Specifically, the gains are most pronounced in Retrieval (including Instruction Retrieval) and Classification (including Pair Classification and Multi-label Classification) tasks, while for Others (including Bitext Mining, Clustering, Reranking, STS and Summariza-

Table 1. Performance comparison between methods across representative models and task types. σ denotes the significance of the improvement.

Model	Method	Retrieval	Classification	Other
Snowflake/snowflake-arctic-embed-m	R1	$+7.15\%$ $73.6\sigma \uparrow$	$+0.62\%$ $7.5\sigma \uparrow$	$+1.39\%$ $4.2\sigma \uparrow$
	R2	$+8.68\%$ $89.4\sigma \uparrow$	$+0.76\%$ $9.0\sigma \uparrow$	$+1.32\%$ $4.0\sigma \uparrow$
sergeyzh/rubert-tiny-turbo	R1	$+1.25\%$ $15.3\sigma \uparrow$	$+0.45\%$ $5.6\sigma \uparrow$	$+0.32\%$ $0.9\sigma \uparrow$
	R2	$+1.28\%$ $15.7\sigma \uparrow$	$+0.46\%$ $5.7\sigma \uparrow$	$+0.41\%$ $1.2\sigma \uparrow$
intfloat/multilingual-e5-large-instruct	R1	$+1.77\%$ $12.7\sigma \uparrow$	$+0.44\%$ $5.6\sigma \uparrow$	$+0.22\%$ $0.7\sigma \uparrow$
	R2	$+1.78\%$ $12.8\sigma \uparrow$	$+0.43\%$ $5.4\sigma \uparrow$	$+0.29\%$ $0.9\sigma \uparrow$
BAAI/bge-base-en	R1	$+1.02\%$ $8.1\sigma \uparrow$	$+0.21\%$ $2.5\sigma \uparrow$	$+0.06\%$ $0.2\sigma \uparrow$
	R2	$+1.05\%$ $8.4\sigma \uparrow$	$+0.22\%$ $2.6\sigma \uparrow$	$+0.06\%$ $0.2\sigma \uparrow$
avsolatorio/GIST-all-MiniLM-L6-v2	R1	$+0.52\%$ $4.0\sigma \uparrow$	$+0.18\%$ $2.2\sigma \uparrow$	$+0.05\%$ $0.1\sigma \uparrow$
	R2	$+0.53\%$ $4.1\sigma \uparrow$	$+0.18\%$ $2.1\sigma \uparrow$	$+0.12\%$ $0.3\sigma \uparrow$
cointegrated/rubert-tiny2	R1	$+0.33\%$ $5.0\sigma \uparrow$	$+0.13\%$ $1.6\sigma \uparrow$	-0.26% $0.8\sigma \downarrow$
	R2	$+0.40\%$ $6.0\sigma \uparrow$	$+0.15\%$ $1.9\sigma \uparrow$	-0.19% $0.6\sigma \downarrow$

Table 2. Comparison of R1 and R2 methods across task types. Sig. means significance.

Task Type	#model	#task	sig. of R1	sig. of R2
Retrieval	38	182	$8.1\sigma \uparrow$	$9.7\sigma \uparrow$
Classification	38	323	$2.9\sigma \uparrow$	$3.1\sigma \uparrow$
Other	38	119	$0.7\sigma \uparrow$	$0.8\sigma \uparrow$

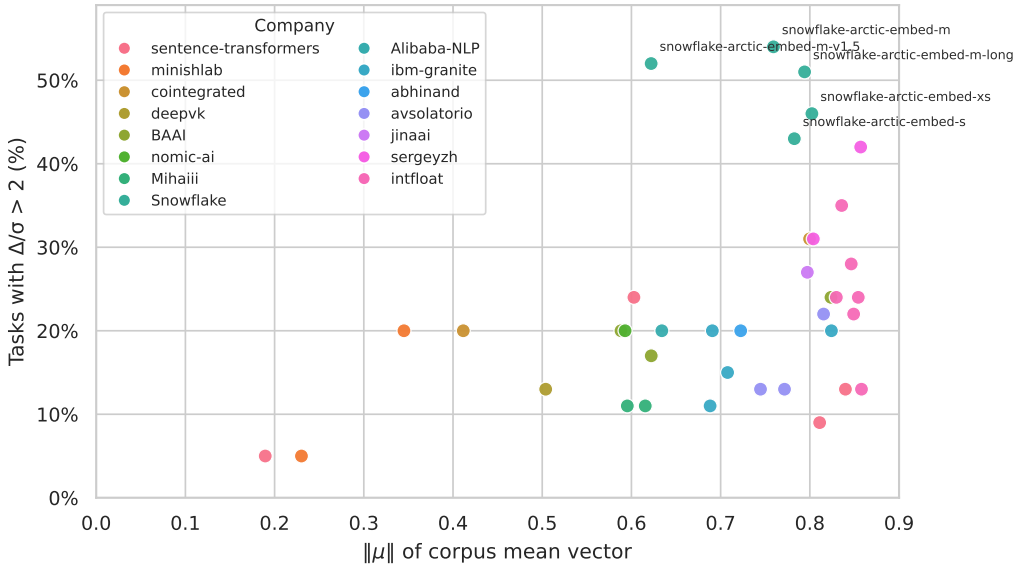

Figure 2. Correlation between model mean vector norm and renormalization effectiveness. The y-axis is the proportion of tasks with significant improvements $> 2\sigma$ for each model in Method R2. We can see that the effectiveness of renormalization has a positive correlation with $\|\mu\|$.

tion) type of tasks, renormalization yields a modest positive effect[2].

Table 2 presents a comparison of the R1 and R2 renormalization methods across different task types, aggregated over 38 models. Both methods yield substantial improvements in retrieval and classification tasks, with smaller but still positive effects on other task types. The improvement is most prominent for retrieval, reaching 9.7σ for R2 and 8.1σ for R1. Overall, R2 consistently outperforms R1.

Figure 1 compares the effects of the two renormalization methods across models in greater detail, with each pair of violin plots representing the signed performance change for a single model, sorted by the mean vector norm $\|\mu\|$. Most results are positive, indicating consistent gains from renormalization. An interesting exception is the model `minishlab/potion-multilingual-128M`, which shows a negative effect under Method R1 but turns positive under Method R2. Notably, Method R2 yields no negative results across all evaluated models, suggesting its greater robustness and stability.

Table 3 summarizes, for each evaluated model, the proportion of tasks that show significant improvement or degradation after applying renormalization[3]. Most models exhibit substantial gains across a majority of tasks, while only a small fraction of tasks experience noticeable declines. To complement this model-level view, we next analyze the same phenomenon from the perspective of individual tasks. Table 4 presents, for each task, the proportion of models that exhibit significant improvement or degradation after applying renormalization. Most retrieval, clustering, and bitext mining benchmarks show broad improvements across a significant fraction of models. Tasks involving reranking or multilabel classification display smaller but still positive effects, while a few individual tasks show limited or mixed responses. This widespread pattern of improvement confirms that renormalization provides robust benefits across diverse architectures and training paradigms.

Figure 2 shows the relationship between the mean vector norm of each model and the effectiveness of renormalization, measured by the proportion of tasks with improvements greater than 2σ. A clear positive correlation can be observed between the two.

More detailed analysis is given in Appendix C. Notably in Table 8, we find that many tasks show very significant improvements, while nearly no task shows significant degrades. For example, the task `WinoGrande` gets a maximum score gain of 0.5559 (the total score ranges from 0 to 1). Given that the renormalization is plug-and-play, training-free and lightweight, it can always be applied to text embedding models to see whether the performance gets better for a particular task.

[2]Most of the task types take a score value from 0 to 1, while some of them take a score value from -1 to 1.

[3]Significance is defined as a change exceeding 2σ. Only tasks providing the sample size n, which allows σ to be computed, are included in this analysis.

3.3. Compute

All experiments were run on a single machine equipped with a Ryzen Threadripper PRO 5975WX CPU and several NVIDIA A6000 GPUs. Typically, completing the filtered MMTEB evaluation for a model of size `100MB` requires `24` hours on a single NVIDIA A6000 GPU.

4. Discussion

Why does classification benefit? Classification tasks depend on clear decision boundaries between classes, which require embeddings with high angular resolution and sufficient separation between neighboring samples. Reducing anisotropy helps restore these properties by spreading sentence embeddings more evenly across the unit sphere. As a result, the embeddings become more linearly separable and the nearest-neighbor margins increase, leading to improved performance for both linear and kNN-based classification models.

Why does retrieval benefit so much? Retrieval tasks involve finding the most relevant items from a large collection given a query. In this sense, retrieval can be viewed as an extreme form of classification, where each possible document or sentence represents a distinct class. Therefore, retrieval tasks are more sensitive to noise, since any perturbation of a single item in a large collection can potentially change the retrieval result. By removing irrelevant directions in the text embedding space, renormalization reduces such noise, which explains why retrieval benefits the most from our method.

5. Related Work

Pre-trained sentence embeddings are known to suffer from distributional degeneration, where representations collapse into a narrow cone that reflects a shared dominant direction. Early work (Arora et al., 2017) showed that removing the first principal component after aggregating word embeddings improves semantic similarity, suggesting that a common component can hinder expressiveness. Subsequent efforts explored ways to alleviate such shared biases. Whitening-based methods (Huang et al., 2021; Su et al., 2021) decorrelate embedding dimensions to obtain more isotropic representations and yield performance gains without supervision. (Fuster Baggetto & Fresno, 2022) pointed out that systematic biases, such as token frequency and

Table 3. Proportion of tasks with significant improvements or degradations ($> 2\sigma$) for each model in Method R2.

Model	$> 2\sigma$ (%)	$< -2\sigma$ (%)
Snowflake/snowflake-arctic-embed-m	54	2
Snowflake/snowflake-arctic-embed-m-v1.5	52	4
Snowflake/snowflake-arctic-embed-m-long	51	0
Snowflake/snowflake-arctic-embed-xs	46	7
Snowflake/snowflake-arctic-embed-s	43	4
sergeyzh/rubert-tiny-turbo	42	9
intfloat/e5-base	35	11
sergeyzh/LaBSE-ru-turbo	31	9
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	24	2
minishlab/potion-multilingual-128M	20	0
cointegrated/rubert-tiny2	31	13
jinaai/jina-embeddings-v2-base-en	27	7
intfloat/e5-small-v2	28	11
cointegrated/LaBSE-en-ru	20	7
nomic-ai/nomic-embed-text-v1	20	5
intfloat/e5-base-v2	24	12
ibm-granite/granite-embedding-125m-english	20	11
BAAI/bge-base-en	24	16
BAAI/bge-base-en-v1.5	20	13
ibm-granite/granite-embedding-30m-english	20	13
Alibaba-NLP/gte-multilingual-base	20	13
abhinand/MedEmbed-small-v0.1	20	15
deepvk/USER-base	13	9
avsolatorio/GIST-all-MiniLM-L6-v2	22	17
sentence-transformers/sentence-t5-large	9	7
sentence-transformers/all-MiniLM-L6-v2	5	2
minishlab/potion-base-8M	5	2
BAAI/bge-small-en-v1.5	17	15

Table 4. Proportion of models showing significant improvements or degradations ($> 2\sigma$) for each task in Method R2.

Task	Type	$> 2\sigma$ (%)	$< -2\sigma$ (%)
BelebeleRetrieval	Retrieval	84	0
TwentyNewsgroupsClusteri	Clustering	76	0
NTREXBitextMining	BitextMining	82	11
STS17	STS	68	11
BUCC.v2	BitextMining	53	5
Touche2020Retrieval.v3	Retrieval	68	24
RedditClusteringP2P.v2	Clustering	54	8
SyntheticText2SQL	Retrieval	55	21
MedrxivClusteringS2S.v2	Clustering	29	3
MedrxivClusteringP2P.v2	Clustering	29	3
IN22GenBitextMining	BitextMining	37	11
TwitterURLCorpus	PairClassifi	29	8
CodeSearchNetRetrieval	Retrieval	25	3
NFCorpus	Retrieval	18	0
AppsRetrieval	Retrieval	18	0
VieMedEVBitextMining	BitextMining	21	3
ClusTREC-Covid	Clustering	18	0
WikipediaRerankingMultil	Reranking	21	3
IndicGenBenchFloresBitex	BitextMining	18	0
LanguageClassification	Classificati	16	0
CodeTransOceanContest	Retrieval	8	0
ESCIReranking	Reranking	9	0
AutoRAGRetrieval	Retrieval	5	0
BornholmBitextMining	BitextMining	5	0
JaqketRetrieval	Retrieval	16	10
ArXivHierarchicalCluster	Clustering	3	0
TbilisiCityHallBitextMin	BitextMining	5	3
CEDRClassification	MultilabelCl	3	0

punctuation, rather than anisotropy alone, degrade semantic quality, and that removing these biases can be beneficial. However, (Mickus et al., 2024) showed that enforcing full isotropy may be sub-optimal, since a certain degree of anisotropy is necessary to preserve cluster structure for classification.

Several training-free or nearly training-free post-processing techniques have been explored to improve sentence embeddings. BERT-flow (Li et al., 2020) applies a normalizing-flow-based transformation to map embeddings into a Gaussian distribution, yielding improvements on semantic textual similarity tasks without task-specific supervision. Ditto (Chen et al., 2023) reweights token contributions based on attention statistics to down-weight uninformative tokens, which mitigates anisotropy and improves semantic similarity without additional training. In addition, (Takeshita et al., 2025) showed that a large portion of embedding dimensions are redundant or even detrimental, and that removing a substantial number of dimensions results in minimal performance degradation across a range of models and tasks, indicating structural inefficiencies in the embedding space. (Thirukovalluru & Dhingra, 2025) constructed embeddings by aggregating multiple paraphrased variants of the same input generated by a large language model, yielding improvements in semantic similarity as well as clustering, reranking, and pairwise classification tasks.

6. Conclusion

We showed that text embedding models exhibit a consistent mean bias and proposed a simple, training-free fix called *renormalization*. It removes the shared mean vector or its projection and consistently improves performance on MMTEB, especially for retrieval and classification tasks. Some particular tasks even get very significant improvements. The variant that removes the projection (R2) performs best, matching our theoretical analysis. Renormalization is lightweight, model-agnostic, and can be directly applied to existing systems, offering an effective way to reduce embedding anisotropy and thus improve model performance efficiently and economically.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Arora, S., Liang, Y., and Ma, T. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR) Workshop*, 2017. URL https://openreview.net/forum?id=SyK00v5xx.

Chen, Q., Wang, W., Zhang, Q., Zheng, S., Deng, C., Yu, H., Liu, J., Ma, Y., and Zhang, C. Ditto: Simple and efficient approach to sentence embeddings. In *Findings of the Association for Computational Linguistics: ACL*, pp. 880–895, 2023. doi: 10.18653/v1/2023.findings-acl.55. URL https://aclanthology.org/2023.findings-acl.55.

Enevoldsen, K., Chung, I., Kerboua, I., Kardos, M., Mathur, A., Stap, D., and et al., J. G. Mmteb: Massive multilingual text embedding benchmark, 2025. URL https://arxiv.org/abs/2502.13595.

Fuster Baggetto, A. and Fresno, V. Is anisotropy really the cause of bert embeddings not being semantic? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8245–8258, 2022. doi: 10.18653/v1/2022.acl-long.566. URL https://aclanthology.org/2022.acl-long.566.

Huang, J., Tang, D., Zhong, W., Lu, S., Shou, L., Gong, M., Jiang, D., and Duan, N. Whiteningbert: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 551–561, 2021. doi: 10.18653/v1/2021.findings-emnlp.48. URL https://aclanthology.org/2021.findings-emnlp.48.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9119–9130, 2020. doi: 10.18653/v1/2020.emnlp-main.733. URL https://aclanthology.org/2020.emnlp-main.733.

Li, X., Zhang, W., Liu, Y., Hu, Z., Zhang, B., and Hu, X. Language-driven anchors for zero-shot adversarial robustness, 2024. URL https://arxiv.org/abs/2301.13096.

Liang, H., Sun, Y., Cai, Y., Zhu, J., and Zhang, B. Jailbreaking llms' safeguard with universal magic words for text embedding models, 2025. URL https://arxiv.org/abs/2501.18280.

Mickus, T., Grönroos, S.-A., and Attieh, J. Isotropy, clusters, and classifiers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. doi: 10.18653/v1/2024.acl-long.XXX. URL https://arxiv.org/abs/2402.09371.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. Mteb: Massive text embedding benchmark, 2023. URL https://arxiv.org/abs/2210.07316.

Su, J., Cao, J., Liu, W., and Ou, Y. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021. URL https://arxiv.org/abs/2103.15316.

Takeshita, S., Takeshita, Y., Ruffinelli, D., and Ponzetto, S. P. Randomly removing 50% of dimensions in text embeddings has minimal impact on retrieval and classification tasks. *arXiv preprint arXiv:2502.01867*, 2025. URL https://arxiv.org/abs/2502.01867.

Thirukovalluru, R. and Dhingra, B. Geneol: Harnessing the generative power of llms for training-free sentence embeddings. *arXiv preprint arXiv:2501.06780*, 2025. URL https://arxiv.org/abs/2501.06780.

A. Filtered Tasks

A total of 192 tasks are excluded from the MMTEB evaluation (version 1.38.18) for reproducibility and fairness. As summarized in Table 5, these removals fall into three categories

1. *Failures*, referring to tasks that consistently crash or produce invalid outputs across multiple runs;
2. *Timeouts*, representing tasks that exceed the predefined wall-clock limit even after retries; and
3. *Unsupported modality*, covering image-based or multimodal benchmarks incompatible with text-only embedding models.

This filtering ensures that all reported results are based on successfully and meaningfully completed evaluations. We maintain a persistent JSON tracker with disjoint categories. Any task previously recorded in these categories is automatically skipped in subsequent runs unless explicitly re-enabled.

Table 5. Tasks filtered out of the MMTEB evaluation.

oprule Category	Count	Tasks
Failures	115	VisualSTS17Eng, VisualSTS17Multilingual, VisualSTS-b-Eng, VisualSTS-b-Multilingual, CodeEditSearchRetrieval, CodeRAGProgrammingSolutions, CodeRAGOnlineTutorials, CodeRAGLibraryDocumentationSolutions, DanFeverRetrieval, TwitterHjerneRetrieval, GreekCivicsQA, BrightRetrieval, BrightLongRetrieval, ClimateFEVER.v2, HagridRetrieval, LEMBNeedleRetrieval, LEMBPasskeyRetrieval, LEMBSummScreenFDRetrieval, MSMARCO, NanoArguAnaRetrieval, NanoClimateFeverRetrieval, NanoDBPediaRetrieval, NanoFEVERRetrieval, NanoFiQA2018Retrieval, NanoHotpotQARetrieval, NanoMSMARCORetrieval, NanoNFCorpusRetrieval, NanoNQRetrieval, NanoQuoraRetrieval, NanoSCIDOCSRetrieval, NanoSciFactRetrieval, NanoTouche2020Retrieval, TopiOCQA, TopiOCQAHardNegatives, MSMARCO-Fa, JaQuADRetrieval, Ko-StrategyQA, CUREv1, MIRACLRetrievalHardNegatives, NeuCLIR2022Retrieval, NeuCLIR2023Retrieval, WebFAQRetrieval, XQuADRetrieval, mMARCO-NL, Touche2020-NL, SNLRetrieval, SpanishPassageRetrievalS2P, SpanishPassageRetrievalS2S, VieQuADRetrieval, T2Retrieval, MMarcoRetrieval, DuRetrieval, CovidRetrieval, CmedqaRetrieval, EcomRetrieval, MedicalRetrieval, VideoRetrieval, WebQAT2TRetrieval, DKHateClassification, FinancialPhrasebankClassification, TweetTopicSingleClassification, DigikalamagClassification, FrenchBookReviews, HindiDiscourseClassification, SentimentAnalysisHindi, IndonesianIdClickbaitClassification, KannadaNewsClassification, KLUE-TC, KorFin, AfriSentiLangClassification, TurkicClassification, MyanmarNews, HateSpeechPortugueseClassification, SanskritShlokasClassification, SinhalaNewsClassification, SinhalaNewsSourceClassification, SpanishNewsClassification, SiswatiNewsClassification, SwahiliNewsClassification, UrduRomanSentimentClassification, IsiZuluNewsClassification, BibleNLPBitextMining, FloresBitextMining, IWSLT2017BitextMining, LinceMTBitextMining, NollySentiBitextMining, NusaTranslationBitextMining, NusaXBitextMining, PhincBitextMining, WebFAQBitextMiningQuestions, WebFAQBitextMiningQAs, DigikalamagClustering, NLPTwitterAnalysisClustering, SNLHierarchicalClusteringP2P, SNLHierarchicalClusteringS2S, SwednClusteringP2P, SwednClusteringS2S, PubChemSMILESPC, indonli, KLUE-NLI, TERRa, Ocnli, Cmnli, WebLINXCandidatesReranking, MIRACLReranking, T2Reranking, MMarcoReranking, CPUSpeedTask, GPUSpeedTask, FaroeseSTS, JSTS, KLUE-STS, MLSUMClusteringP2P.v2, SIB200ClusteringS2S, WikiClusteringP2P.v2
Timeouts	28	CodeRAGStackoverflowPosts, MSMARCOv2, NQ, ClimateFEVER-Fa, DBPedia-Fa, HotpotQA-Fa, NQ-Fa, MIRACLRetrieval, MrTidyRetrieval, MultiLongDocRetrieval, ClimateFEVER-NL, DBPedia-NL, FEVER-NL, HotpotQA-NL, NQ-NL, DBPedia-PL, HotpotQA-PL, MSMARCO-PL, NQ-PL, ClimateFEVER, DBPedia, FEVER, HotpotQA, RiaNewsRetrieval, mFollowIRCrossLingualInstructionRetrieval, mFollowIRInstructionRetrieval, MultiEURLEXMultilabelClassification, GerDaLIR
Unsupported modality	49	CUB200I2IRetrieval, FORBI2IRetrieval, GLDv2I2IRetrieval, METI2IRetrieval, NIGHTSI2IRetrieval, ROxfordEasyI2IRetrieval, ROxfordMediumI2IRetrieval, ROxfordHardI2IRetrieval, RP2kI2IRetrieval, RParisEasyI2IRetrieval, RParisMediumI2IRetrieval, RParisHardI2IRetrieval, SketchyI2IRetrieval, SOPI2IRetrieval, StanfordCarsI2IRetrieval, Birdsnap, Caltech101, CIFAR10, CIFAR100, Country211, DTD, EuroSAT, FER2013, FGVCAircraft, Food101Classification, GTSRB, Imagenet1k, MNIST, OxfordFlowersClassification, OxfordPets, PatchCamelyon, RESISC45, StanfordCars, STL10, SUN397, UCF101, CIFAR10Clustering, CIFAR100Clustering, ImageNetDog15Clustering, ImageNet10Clustering, TinyImageNetClustering, VOC2007, STS12VisualSTS, STS13VisualSTS, STS14VisualSTS, STS15VisualSTS, STS16VisualSTS, STS17MultilingualVisualSTS, STSBenchmarkMultilingualVisualSTS

B. Evaluated Models

Table 6 lists all embedding models evaluated in this study. The selection follows the official MMTEB leaderboard, covering a broad range of widely used multilingual and English models from major organizations. For each model, we report its model size (*Size(MB)*), mean vector norm (*Mean*), and the corresponding leaderboard rank (*MTEB rank*), providing a concise overview of their scale and relative performance.

Table 6. MMTEB models evaluated in this study, sorted by official leaderboard rank.

Model	Size(MB)	Mean	MTEB rank
intfloat/multilingual-e5-large-instruct	1068	0.8491	7
Alibaba-NLP/gte-multilingual-base	582	0.6341	26
intfloat/multilingual-e5-small	449	0.8543	38
ibm-granite/granite-embedding-278m-multilingual	530	0.6882	44
ibm-granite/granite-embedding-107m-multilingual	204	0.7078	52
nomic-ai/nomic-embed-text-v1	522	0.5929	57
intfloat/e5-base-v2	418	0.8297	59
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	449	0.6028	64
avsolatorio/GIST-Embedding-v0	418	0.7447	67
BAAI/bge-base-en-v1.5	390	0.5883	75
avsolatorio/GIST-small-Embedding-v0	127	0.7716	76
minishlab/potion-multilingual-128M	489	0.3450	79
intfloat/e5-small-v2	127	0.8464	82
intfloat/e5-base	418	0.8357	83
BAAI/bge-small-en-v1.5	127	0.6222	86
abhinand/MedEmbed-small-v0.1	127	0.7225	91
ibm-granite/granite-embedding-125m-english	238	0.8242	92
intfloat/e5-small	127	0.8579	93
Snowflake/snowflake-arctic-embed-m-long	522	0.7941	98
avsolatorio/GIST-all-MiniLM-L6-v2	87	0.8154	100
sergeyzh/LaBSE-ru-turbo	490	0.8039	101
Snowflake/snowflake-arctic-embed-m	415	0.7592	103
Snowflake/snowflake-arctic-embed-s	127	0.7826	104
Snowflake/snowflake-arctic-embed-m-v1.5	415	0.6221	107
cointegrated/LaBSE-en-ru	492	0.4115	108
ibm-granite/granite-embedding-30m-english	58	0.6907	110
sentence-transformers/all-MiniLM-L6-v2	87	0.1895	117
Mihaiii/Wartortle	66	0.6155	118
deepvk/USER-base	473	0.5039	120
Snowflake/snowflake-arctic-embed-xs	86	0.8023	124
Mihaiii/Squirtle	60	0.5954	128
sergeyzh/rubert-tiny-turbo	111	0.8570	130
minishlab/potion-base-8M	29	0.2301	131
cointegrated/rubert-tiny2	112	0.7997	138
jinaai/jina-embeddings-v2-base-en	262	0.7972	154
sentence-transformers/sentence-t5-large	639	0.8110	179
BAAI/bge-base-en	390	0.8237	189
sentence-transformers/sentence-t5-base	209	0.8399	192

C. Extended Tables and Figures

Table 7 summarizes the per-model performance changes under Method R2, focusing on tasks with substantial deviations after renormalization, defined by thresholds of $|\Delta| > 0.1$ (score difference) and relative change $|\delta| > 2\%$ (percentage of the score difference compared to the original score). For each model, we report the number of tasks with significant improvement, the average score gain across those improved tasks, the largest gain, and the smallest gain. We also report the number of tasks with significant degradation, the average score drop across those degraded tasks, the largest drop, and the smallest drop.

Similarly, Table 8 presents task-level statistics under Method R2, where each row corresponds to a single MMTEB task. For each task, we report the number of models that show significant improvement, along with the average, maximum, and minimum score gains across those models. The same information is provided for models exhibiting performance degradation, including the number of affected models and the corresponding average, maximum, and minimum score drops.

Like Figure 2, Figure 3 explores the relationship between the strength of renormalization effects and the model's mean vector norm. The difference is that Figure 2 measures effectiveness by the proportion of tasks showing significant improvement, whereas Figure 3 plots the average improvement Δ across all tasks.

Table 7. Performance of Method R2: Significant Task Improvements and Degradations. The filter is that $|\Delta| > 0.1$ and $|\delta| > 2\%$, where Δ is the score difference and δ is the percentage of the score difference compared to the original score.

Model	$\# \uparrow$	$\bar{\Delta}_{\uparrow}$	$\Delta_{\max,\uparrow}$	$\Delta_{\min,\uparrow}$	$\# \downarrow$	$\bar{\Delta}_{\downarrow}$	$\Delta_{\max,\downarrow}$	$\Delta_{\min,\downarrow}$
Snowflake/snowflake-arctic-embed-m	59	0.2346	0.5849	0.1019	3	-0.1462	-0.1264	-0.1589
Snowflake/snowflake-arctic-embed-m-v1.5	53	0.1908	0.5338	0.1011	2	-0.1058	-0.1034	-0.1081
Snowflake/snowflake-arctic-embed-s	43	0.1625	0.5559	0.1021	0	–	–	–
Snowflake/snowflake-arctic-embed-m-long	39	0.1905	0.3992	0.1022	2	-0.1341	-0.1149	-0.1532
Snowflake/snowflake-arctic-embed-xs	29	0.1536	0.3092	0.1011	1	-0.1682	-0.1682	-0.1682
intfloat/multilingual-e5-large-instruct	4	0.1192	0.1407	0.1082	1	-0.1308	-0.1308	-0.1308
sergeyzh/LaBSE-ru-turbo	3	0.1982	0.3586	0.1150	1	-0.1589	-0.1589	-0.1589
sergeyzh/rubert-tiny-turbo	3	0.1573	0.2028	0.1102	1	-0.1215	-0.1215	-0.1215
intfloat/e5-base	2	0.1489	0.1871	0.1108	0	–	–	–
intfloat/e5-base-v2	2	0.1740	0.2289	0.1191	0	–	–	–
nomic-ai/nomic-embed-text-v1	2	0.1148	0.1271	0.1025	0	–	–	–
cointegrated/rubert-tiny2	2	0.1362	0.1553	0.1171	1	-0.1192	-0.1192	-0.1192
sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2	2	0.1034	0.1040	0.1028	1	-0.2316	-0.2316	-0.2316
BAAI/bge-small-en-v1.5	1	0.1130	0.1130	0.1130	0	–	–	–
ibm-granite/granite-embedding-107m-multilingual	1	0.1723	0.1723	0.1723	0	–	–	–
intfloat/e5-small	1	0.1709	0.1709	0.1709	0	–	–	–
intfloat/e5-small-v2	1	0.1729	0.1729	0.1729	0	–	–	–
sentence-transformers/sentence-t5-large	1	0.1227	0.1227	0.1227	0	–	–	–
BAAI/bge-base-en	0	–	–	–	1	-0.1669	-0.1669	-0.1669
ibm-granite/granite-embedding-125m-english	0	–	–	–	1	-0.1101	-0.1101	-0.1101
intfloat/multilingual-e5-small	0	–	–	–	1	-0.1215	-0.1215	-0.1215
jinaai/jina-embeddings-v2-base-en	0	–	–	–	1	-0.1028	-0.1028	-0.1028

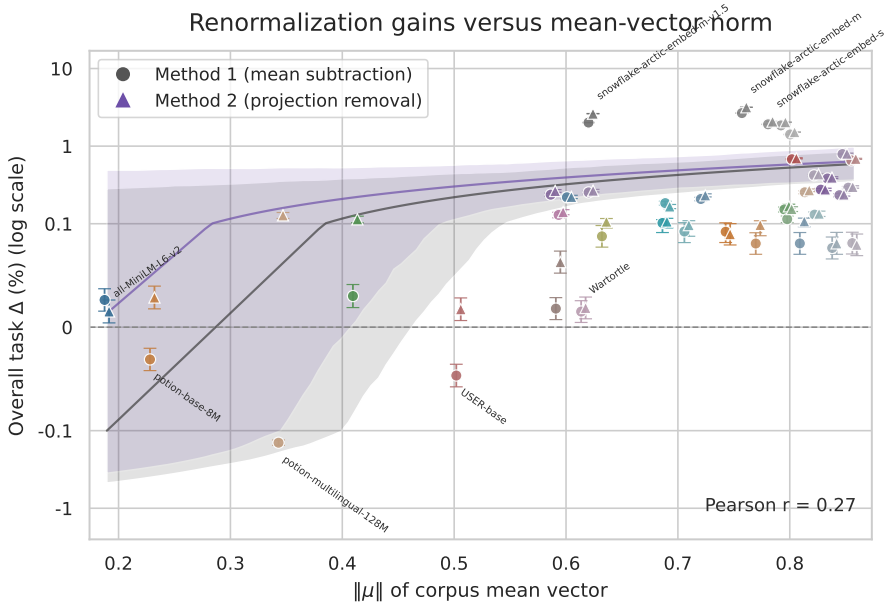

Figure 3. Correlation between model mean vector norm and renormalization effectiveness. The y-axis is the task score difference Δ properly scaled by logarithm and the error bar is the sigma calculated in Table 3. We try to fit these scattered points by the line and shadow to see that these two are positively related.

Table 8. Performance of Method R2: Significant Improvements and Degradations. The filter is that $|\Delta| > 0.1$ and $|\delta| > 2\%$, where Δ is the score difference and δ is the percentage of the score difference compared to the original score. N is the total number of models, which is split into the number of models getting better and getting worse.

Task	N	$\#\uparrow$	$\bar{\Delta}_{\uparrow}$	$\Delta_{\max,\uparrow}$	$\Delta_{\min,\uparrow}$	$\#\downarrow$	$\bar{\Delta}_{\downarrow}$	$\Delta_{\max,\downarrow}$	$\Delta_{\min,\downarrow}$
WinoGrande	10	10	0.2121	0.5559	0.1150	0	–	–	–
TextualismToolDictionari	8	1	0.1028	0.1028	0.1028	7	-0.1375	-0.1028	-0.1682
ChemHotpotQARetrieval	6	5	0.4234	0.5849	0.2728	1	-0.1192	-0.1192	-0.1192
WikipediaSpecialtiesInCh	6	5	0.1525	0.2289	0.1130	1	-0.1669	-0.1669	-0.1669
CQADupstackEnglishRetrie	5	5	0.1562	0.2108	0.1240	0	–	–	–
CQADupstackGamingRetriev	5	5	0.1963	0.3450	0.1295	0	–	–	–
CQADupstackPhysicsRetrie	5	5	0.1772	0.2824	0.1180	0	–	–	–
CQADupstackProgrammersRe	5	5	0.1544	0.2198	0.1321	0	–	–	–
CQADupstackRetrieval	5	5	0.1508	0.2450	0.1099	0	–	–	–
CQADupstackTexRetrieval	5	5	0.1320	0.1772	0.1059	0	–	–	–
CQADupstackUnixRetrieval	5	5	0.1448	0.2557	0.1080	0	–	–	–
CQADupstackWordpressRetr	5	5	0.1491	0.2460	0.1093	0	–	–	–
ChemNQRetrieval	5	5	0.2360	0.2884	0.1785	0	–	–	–
FEVERHardNegatives	5	5	0.3113	0.5009	0.1407	0	–	–	–
FeedbackQARetrieval	5	5	0.2353	0.3549	0.1270	0	–	–	–
FiQA2018	5	5	0.1620	0.1944	0.1315	0	–	–	–
HotpotQA-PLHardNegatives	5	5	0.1436	0.1743	0.1011	0	–	–	–
HotpotQAHardNegatives	5	5	0.3459	0.4562	0.2573	0	–	–	–
LitSearchRetrieval	5	5	0.3329	0.3992	0.2690	0	–	–	–
MSMARCOHardNegatives	5	5	0.1435	0.1830	0.1223	0	–	–	–
NQHardNegatives	5	5	0.2532	0.3002	0.1909	0	–	–	–
SyntecRetrieval	5	5	0.2100	0.3544	0.1046	0	–	–	–
SyntheticText2SQL	5	5	0.1834	0.2605	0.1254	0	–	–	–
Touche2020Retrieval.v3	5	5	0.2245	0.2698	0.1903	0	–	–	–
CQADupstackGisRetrieval	4	4	0.1655	0.2725	0.1039	0	–	–	–
CQADupstackMathematicaRe	4	4	0.1384	0.1854	0.1154	0	–	–	–
CQADupstackStatsRetrieva	4	4	0.1485	0.1972	0.1244	0	–	–	–
CUADThirdPartyBeneficiar	4	4	0.1434	0.1618	0.1029	0	–	–	–
DBPediaHardNegatives	4	4	0.1995	0.2422	0.1471	0	–	–	–
LEMBWikimQARetrieval	4	4	0.1961	0.3208	0.1479	0	–	–	–
LegalBenchCorporateLobby	4	4	0.2605	0.5331	0.1023	0	–	–	–
MLQuestions	4	4	0.1694	0.2086	0.1163	0	–	–	–
MedicalQARetrieval	4	4	0.2924	0.3584	0.1618	0	–	–	–
NFCorpus	4	4	0.2106	0.2569	0.1299	0	–	–	–
SCIDOCS	4	4	0.1327	0.1471	0.1154	0	–	–	–
SciFact	4	4	0.3423	0.5767	0.1504	0	–	–	–
BSARDRetrieval	3	3	0.1276	0.1441	0.1036	0	–	–	–
BuiltBenchRetrieval	3	3	0.2456	0.4602	0.1104	0	–	–	–
CQADupstackAndroidRetrie	3	3	0.1811	0.2940	0.1022	0	–	–	–
CQADupstackWebmastersRet	3	3	0.1689	0.2543	0.1090	0	–	–	–
ClimateFEVERHardNegative	3	3	0.1654	0.2016	0.1128	0	–	–	–
FQuADRetrieval	3	3	0.2077	0.2391	0.1705	0	–	–	–
GermanDPR	3	3	0.1210	0.1442	0.1093	0	–	–	–
GermanGovServiceRetrieva	3	3	0.1582	0.2028	0.1040	0	–	–	–
PubChemWikiPairClassific	3	3	0.1681	0.2348	0.1213	0	–	–	–
SKQuadRetrieval	3	3	0.1651	0.1768	0.1424	0	–	–	–
SlovakSumRetrieval	3	3	0.1336	0.1577	0.1019	0	–	–	–
SweFaqRetrieval	3	3	0.1139	0.1277	0.1011	0	–	–	–
SwednRetrieval	3	3	0.1342	0.1522	0.1202	0	–	–	–
TV2Nordretrieval	3	3	0.1474	0.1705	0.1134	0	–	–	–
ContractNLIPermissibleCo	3	0	–	–	–	3	-0.1149	-0.1034	-0.1264
ContractNLIPermissiblePo	3	0	–	–	–	3	-0.1382	-0.1081	-0.1532
CUADJointIPOwnershipLega	2	2	0.1849	0.2448	0.1250	0	–	–	–
GermanQuAD-Retrieval	2	2	0.1883	0.2039	0.1727	0	–	–	–
LEMBNarrativeQARetrieval	2	2	0.1135	0.1138	0.1132	0	–	–	–
NarrativeQARetrieval	2	2	0.1137	0.1138	0.1136	0	–	–	–
SciFact-NL	2	2	0.1417	0.1558	0.1277	0	–	–	–
SyntecReranking	2	2	0.1537	0.2028	0.1045	0	–	–	–
WikipediaChemistryTopics	2	2	0.1074	0.1108	0.1040	0	–	–	–
WikipediaRetrievalMultil	2	2	0.1333	0.1490	0.1177	0	–	–	–
ZacLegalTextRetrieval	2	2	0.1273	0.1464	0.1082	0	–	–	–
IndicCrosslingualSTS	2	1	0.1209	0.1209	0.1209	1	-0.1101	-0.1101	-0.1101