Geometry-Preserving Dimensionality Reduction for Text Embeddings

Konstantinos Kalyfommatos[1,2] **Andrianos Michail**[2] **Marius Huber**[2] **Bill Psomas**[3]
Simon Clematide[2] **Juri Opitz**[2]

[1]National and Kapodistrian University of Athens [2]University of Zurich
[3]VRG, FEE, Czech Technical University in Prague

Abstract

Dense text embeddings are a core representation in modern NLP, supporting tasks such as retrieval, clustering, classification, and semantic search. However, embeddings often have hundreds or thousands of dimensions, creating substantial storage and efficiency challenges at scale. In this work, we present a systematic study of post-hoc dimensionality reduction methods for text embeddings across multiple modern embedding backbones, compression ratios, and downstream tasks. We introduce GEOPRES, a simple geometry–preserving reduction method: a learned linear map trained to preserve pairwise distances in the original embedding space—motivated by the Johnson–Lindenstrauss lemma from metric geometry. Our experiments show that embedding dimensionality can often be substantially reduced with minimal downstream task performance loss, and that GEOPRES outperforms competing methods across many settings. We further find that preserving internal similarity rankings strongly correlates with downstream utility, providing a useful proxy for evaluating reduction quality. Overall, our results offer practical recommendations for selecting dimensionality reduction techniques in text embedding models.

1 Introduction

Text embeddings are central to modern NLP, providing a shared vector space in which sentences, documents, and queries can be compared. Use cases include semantic search and clustering, topic discovery, recommendation, dataset curation, and retrieval-augmented generation (Opitz et al., 2025). Typically, embedding models are contrastively trained to map text to dense, fixed-dimensional vectors—see, e.g., the pioneering works of Reimers and Gurevych (2019) and Gao et al. (2021) with extension to decoder-based LLMs by Muennighoff (2022) and BehnamGhader et al. (2024).

Due to their usefulness, embeddings are often produced at massive scale. In combination with their high dimensionality, this creates practical pressure points across the stack, including storage and memory footprint, similarity computation, and nearest-neighbor search. For example, simulations with Hierarchical Navigable Small World graphs (HNSW, Malkov and Yashunin, 2018) show that search accuracy and latency deteriorate as dimensionality increases (Figure 1).

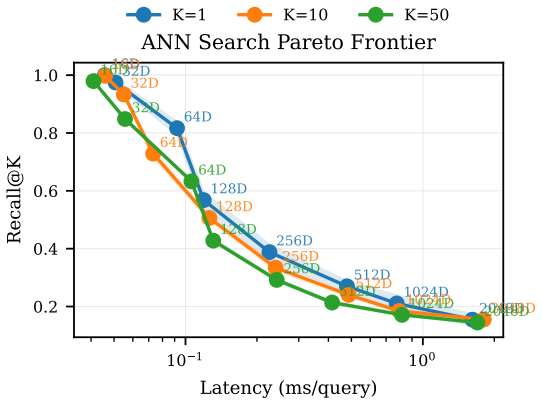


Figure 1: Effect of embedding dimensionality on ANN search. Both query latency and recall@K deteriorate as dimensionality increases, consistent with practical curse-of-dimensionality effects. Results are averaged over five runs with 20,000 points and 300 queries.

This motivates a practical question: *How can embedding dimensionality be reduced while preserving downstream utility?*

In this work, we focus on post-hoc dimensionality reduction methods that are broadly applicable, computationally lightweight, and capable of maintaining strong downstream performance across tasks and embedding models.

The practical relevance of this problem has motivated several related studies. For instance, Takeshita et al. (2025) recently found that randomly removing 50% of dimensions in text embeddings has limited impact on retrieval and classification tasks, suggesting that modern embedding spaces

contain substantial redundancy. However, their reported accuracy loss of "less than 10%" may still be too large for some applications. Random dimension removal is an interesting and strong baseline, but not necessarily a sufficient compression strategy.

Another focal concept is "Matryoshka Representation Learning" (MRL; Kusupati et al. (2022)) that trains with intertwined sub-embedding losses. While highly effective, MRL-based approaches are tightly integrated into backbone training and are limited to models explicitly optimized for this objective; they therefore fall outside a strict post-hoc setting.

Our contributions are fourfold:

1. We present a systematic empirical study of post-hoc dimensionality reduction methods for modern text embeddings across multiple embedding backbones, compression ratios, and downstream tasks.

2. We propose GEOPRES, a lightweight geometry-preserving linear mapping trained to match pairwise structure in embedding spaces, providing a simple and broadly applicable post-hoc reduction method.

3. We analyze the relationship between intrinsic geometric preservation metrics and downstream embedding performance, finding that preserving similarity rankings strongly correlates with task utility.

4. Based on our experimental results, we derive practical recommendations for selecting dimensionality reduction techniques under different deployment scenarios.

Broadly, our study also investigates the relationship between geometric fidelity and downstream embedding utility. We ask to what degree downstream task performance, such as clustering performance, can be predicted by intrinsic metrics that compare the spaces before and after dimensionality reduction. For instance, we ask whether preserving angles is more important than preserving distances or the rankings of internal similarities.

We will release our SentenceTransformer-compatible implementation, trained models, and experimental framework at anonymized_github to facilitate future research.

2 Related Work

Matryoshka Representation Learning (MRL) introduces a nested optimization objective that hierarchically organizes information across embedding dimensions, enabling truncation-based reduction without retraining (Kusupati et al., 2022). While effective, such approaches are tightly coupled to the training process and therefore limited to models explicitly designed for this objective.

From a theoretical perspective, random projections based on the Johnson–Lindenstrauss lemma guarantee approximate distance preservation under dimensionality reduction (Johnson and Lindenstrauss, 1984). Empirical studies demonstrate that such simple, model-agnostic methods can perform competitively in practice (Bingham and Mannila, 2001). Complementing this view, recent work shows that even randomly removing a substantial fraction of embedding dimensions leads to only moderate degradation in downstream performance, further suggesting significant redundancy in modern embedding spaces (Takeshita et al., 2025).

In contrast, established manifold learning techniques such as t-SNE, UMAP, and Isomap construct low-dimensional representations by jointly optimizing over a fixed dataset, preserving local or global geometric structure (McInnes et al., 2018; Tenenbaum et al., 2000). However, their offline nature—i.e., they do not learn an explicit out-of-sample mapping and typically require recomputing the embedding when new points are added—limits their applicability in typical embedding pipelines, where each input is embedded individually rather than jointly with others.

Overall, prior work highlights a trade-off between structural fidelity, computational efficiency, and general applicability, motivating lightweight, post-hoc approaches that can operate on arbitrary embeddings while preserving their geometric and semantic properties.

3 Method

3.1 Formal Research Problem

Given a function $g : X \to Y \subseteq \mathbb{R}^n$, where X is an arbitrary set[1] and n is large, and given $k < n$, we are interested in a function $f^* : Y \to \mathbb{R}^k$, with the property that downstream task performance achieved by $f^* \circ g$ closely matches that of g. With

[1] In the present context, X is a set of sentences, and g maps these sentences into a vector space.

such a function, for any $x \in X$, the embedding $f^*(g(x)) \in \mathbb{R}^k$ serves as a low-dimensional representative of $g(x) \in \mathbb{R}^n$, significantly reducing computational cost without substantial loss in task performance.

3.2 Idea

Our approach is straightforward: we generate a large collection of high-dimensional vectors using g, which serve as training data to approximate f^* with a linear model $f(y) = Wy$, optimized via backpropagation (Section 4.2) using a distance-preserving loss function (Section 3.3). This approach rests on two assumptions: first, that a linear function with the desired properties exists; and second, that preserving intrinsic properties for a large set of vectors in the image of g is predictive of downstream task performance.

Note that PyTorch weight matrices happen to be initialized such that the conclusion of the Johnson–Lindenstrauss lemma applies to the initial projection W; that is, W starts out approximately preserving pairwise distances with high probability.[2] Hence, our method can be seen as "refining" the pairwise distance preservation for the underlying function g.

3.3 Loss Function

We optimize f by minimizing a pairwise distance preservation loss. Formally, let $\mathcal{B} = \{y_1, y_2, \ldots, y_m\} \subseteq \mathbb{R}^n$ be a training batch. For $y_i, y_j \in \mathcal{B}$, let $d_{i,j}$ denote the Euclidean distance between y_i and y_j, and let $d^f_{i,j}$ denote the Euclidean distance between $f(y_i)$ and $f(y_j)$. The loss is defined as

$$\mathcal{L} = \frac{1}{\binom{m}{2}} \sum_{i<j} \left(d_{i,j} - d^f_{i,j} \right)^2 . \qquad (1)$$

Since cosine similarity is the predominant metric in downstream evaluation, one might expect optimizing for cosine similarity preservation to yield better results. Contrary to this expectation, our findings indicate that optimizing for Euclidean distance preservation yields slightly better results, as discussed in Section A.1.1. Experiments with alternative loss functions are discussed in Section A.1.

[2] Linear layers in PyTorch are initialized with weights from a mean-centered uniform distribution, which satisfies the conditions of the Johnson–Lindenstrauss lemma (Vershynin, 2018, Lemma 9.2.4).

4 Experimental Setup

4.1 Backbones

We experiment with four widely used embedding models as backbones, chosen for their practical relevance and computational efficiency. To ensure diverse coverage, two support MRL and two do not.

jinaai/jina-embeddings-v2-small-en (Jina) (Günther et al., 2023) A 33M-parameter English-only model based on a modified BERT architecture (JinaBERT). It produces non–ℓ_2-normalized 512-dimensional embeddings via mean pooling.

sentence-transformers/all-mpnet-base-v2 (MPNet) A 110M-parameter English-only sentence embedding model, fine-tuned on large-scale sentence-pair data, based on MPNet (Song et al., 2020). It produces ℓ_2-normalized 768-dimensional embeddings via mean pooling.

Alibaba-NLP/gte-multilingual-base (mGTE) (Zhang et al., 2024) A 305M-parameter multilingual text encoder (mGTE-TRM). It produces ℓ_2-normalized 768-dimensional embeddings via [CLS] pooling.

Qwen/Qwen3-Embedding-0.6B (Qwen) (Zhang et al., 2025) A 0.6B-parameter decoder-based model built on the Qwen3-0.6B-Base LLM (Team, 2025). Unlike the encoder backbones, it uses last-token pooling, producing ℓ_2-normalized 1024-dimensional embeddings.

MRL support. Among the four models, mGTE and Qwen support MRL; MPNet and Jina do not.

4.2 Training Setup

For each backbone, text embeddings are precomputed for both training and evaluation, as described in Section 3.2.

For training, we randomly sample 10,000,000 text passages from the English subset of the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020), a massive web-crawled corpus. We hypothesize that random sampling is sufficient for our objective: because f is trained only to preserve the intrinsic geometry of a fixed embedding distribution (rather than to model any downstream task or domain), we primarily need broad coverage of "typical" points in the backbone's latent space. Samples from a large, heterogeneous corpus such as C4 provide a broad mixture of topics and styles, making them

suitable for this objective. We also hypothesize that English data alone suffices even for multilingual backbones: since f operates purely on the embedding space rather than on raw text tokens, it is the geometry of that space that we aim to model (we empirically examine this assumption below).

We optimize the parameters using the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 10^{-2} and a weight decay of 0.1. Training is conducted with a large batch size of 20,000 points (drawn uniformly at random from the training set) to ensure that the pairwise distance matrix captures a broad approximation of the global geometric structure in every step. We employ a linear learning rate scheduler with a warmup ratio of 0.1 and train for a total of 10 epochs. Early stopping with a patience of 3 evaluation steps is used to prevent overfitting, restoring the checkpoint with the lowest validation loss.

For validation and testing, we evaluate on two batches of 10,000 paraphrase pairs, ensuring coverage of both similar and unrelated texts. This choice, however, is largely optional: any sufficiently large random sample of texts naturally yields a broad variety of pairwise relationships in the embedding space, as a large heterogeneous corpus inherently spans a wide range of semantic similarities.[3]

4.3 Baselines

We compare our method against the following baselines. Each baseline can be viewed as a head applied on top of a frozen backbone embedding model, mapping high-dimensional embeddings in $\mathbb{R}^n$ to a lower-dimensional space $\mathbb{R}^k$ during inference.

Autoencoder. The autoencoder (Hinton and Salakhutdinov, 2006) is an intuitive baseline for neural dimensionality reduction. It consists of an encoder $f_{\text{enc}} : \mathbb{R}^n \to \mathbb{R}^k$ and a decoder $f_{\text{dec}} : \mathbb{R}^k \to \mathbb{R}^n$, both parameterized as single-layer networks with ReLU activations, trained jointly to minimize the mean squared reconstruction loss $\frac{1}{|\mathcal{B}|} \sum_{x_i \in \mathcal{B}} \|x_i - f_{\text{dec}}(f_{\text{enc}}(x_i))\|_2^2$ (where $\mathcal{B}$ denotes a training batch).

The autoencoder is trained under the same setup as our proposed model, using the same precomputed embeddings, batch size, and number of training steps, ensuring a fair comparison. At inference time, only the encoder f_{enc} is used as the projection head, discarding the decoder. Unlike our method, the autoencoder is trained to minimize reconstruction error in the input space, with no explicit objective to preserve geometric structure in $\mathbb{R}^k$.

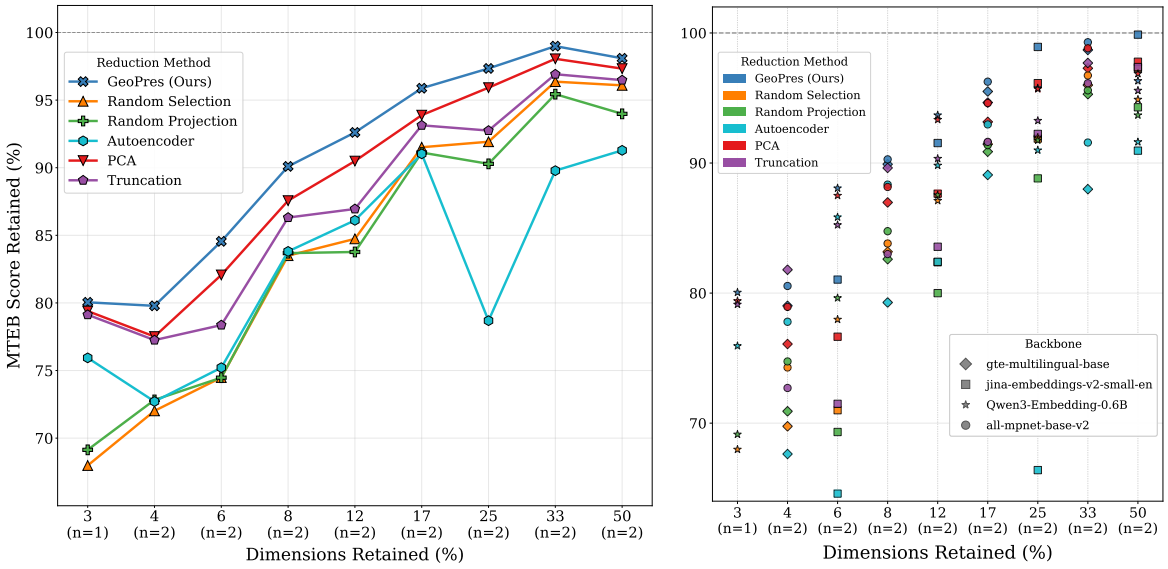
Random Selection. Following Takeshita et al. (2025), random selection uniformly samples a subset $S \subset \{1, \ldots, n\}$ of size k without replacement, and retains only those dimensions: $y_S \in \mathbb{R}^k$. Unlike truncation, the selected dimensions are not necessarily the leading ones. The random subset S is fixed at evaluation time for reproducibility.

Random Projection. A random matrix $\mathbf{W}^{\text{rand}} \in \mathbb{R}^{k \times n}$ is sampled once and then fixed. Its entries are drawn i.i.d. from $\mathcal{N}(0, 1)$, and it is used to project y as $z = \mathbf{W}^{\text{rand}} y \in \mathbb{R}^k$. This construction is motivated by the Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984), which guarantees that pairwise distances are approximately preserved with high probability under $\mathbf{W}^{\text{rand}}$ when $k = \mathcal{O}(\varepsilon^{-2} \log m)$, for a distortion tolerance $\varepsilon > 0$ and m data points. It is thus also an ideal baseline for our approach because our method directly optimizes the projection weights to preserve distances.

Principal Component Analysis (PCA). PCA (Pearson, 1901; Hotelling, 1933) computes an orthonormal basis for $\mathbb{R}^n$ by performing an eigendecomposition of the empirical covariance matrix $\Sigma = \frac{1}{m} Y^\top Y \in \mathbb{R}^{n \times n}$, where $Y \in \mathbb{R}^{m \times n}$ is the matrix of mean-centered training embeddings. The eigenvectors corresponding to the k largest eigenvalues form the projection matrix $W_{\text{PCA}} \in \mathbb{R}^{k \times n}$. At inference time, the vector $y \in \mathbb{R}^n$ is projected as $z = W_{\text{PCA}} y \in \mathbb{R}^k$. In practice, W_{PCA} is estimated from a subset of $m = 300{,}000$ embeddings sampled from the precomputed training set. The choice of 300,000 vectors is intended to provide a stable estimate of the leading principal components. Since all evaluated embedding dimensions satisfy $n \leq 1024$, this sample size is several hundred times larger than the ambient dimension, and is therefore a conservative choice for estimating the empirical covariance structure used by PCA.

Truncation & Matryoshka Representation Learning. Truncation retains only the first k components of each embedding vector, yielding the projection $y_{1:k} \in \mathbb{R}^k$. This is a parameter-free and computationally trivial baseline. For backbones supporting MRL, truncation is the intended

[3] Dataset: https://hf.co/datasets/agentlans/sentence-paraphrases



(a) Each point denotes the average value over the n available backbones.

(b) Each point is one backbone-method–dimension configuration; n denotes the number of backbones per ratio.

Figure 2: Normalized MTEB score retained as a function of the fraction of original embedding dimensions retained.

inference-time mechanism, as MRL explicitly encourages the model to encode coarse-grained information in the leading dimensions and finer-grained information in later ones (Kusupati et al., 2022). For non-MRL backbones, however, no such ordering is guaranteed, and truncation discards potentially informative dimensions arbitrarily.

4.4 Evaluation Tasks and Metric

We evaluate on four general task groups from the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023): STS, Retrieval, Classification, and Clustering. For each task, we report the average of the primary metric over the test split of all included datasets. Following the notation of Section 3, let $g : X \to \mathbb{R}^n$ denote the backbone embedding model and $f : \mathbb{R}^n \to \mathbb{R}^k$ the dimensionality reduction mapping. For a given task metric M, we define the **normalized score** of the composition $f \circ g$ relative to the backbone alone as:

$$\widetilde{M}(f \circ g) = \frac{M(f \circ g)}{M(g)}, \tag{2}$$

where $M(f \circ g)$ denotes the value of the metric obtained using the reduced-dimensional embeddings $f(g(x))$, and $M(g)$ is the value of the metric using the backbone's high-dimensional embeddings $g(x)$. A normalized score of $\widetilde{M}(f \circ g) = 1$ indicates perfect preservation of the original performance, while values below 1 reflect performance degradation. This normalization allows us to directly compare the relative efficacy of different dimensionality reduction methods across diverse tasks and backbones. The score can exceed 1 when the reduced embedding achieves a higher evaluation score than the original backbone embedding.

5 Main Results

Overall NLP Task Performance. At a high level, we ask the following question: *Across all tested backbones, which dimensionality reduction method works best?* Figure 2a illustrates the mean normalized MTEB score retained by each method as a function of the ratio of dimensions kept. The plot confirms that GEOPRES (blue line) consistently achieves the highest average retention across nearly all compression ratios, while PCA remains a competitive and consistent baseline. Random projection, random selection, and the autoencoder underperform, with the latter showing substantial training instability.

To further analyze the results, Figure 2b presents a scatter plot where each point represents a specific configuration of backbone, method, and dimension-retention ratio. Again, the x-axis denotes the ratio of original dimensions retained, while the y-axis shows the per-backbone-normalized MTEB score. This visualization highlights two key insights. First, the superiority of GEOPRES is not driven by a single backbone; blue markers (GEOPRES) consistently occupy the upper envelope of the distribu-

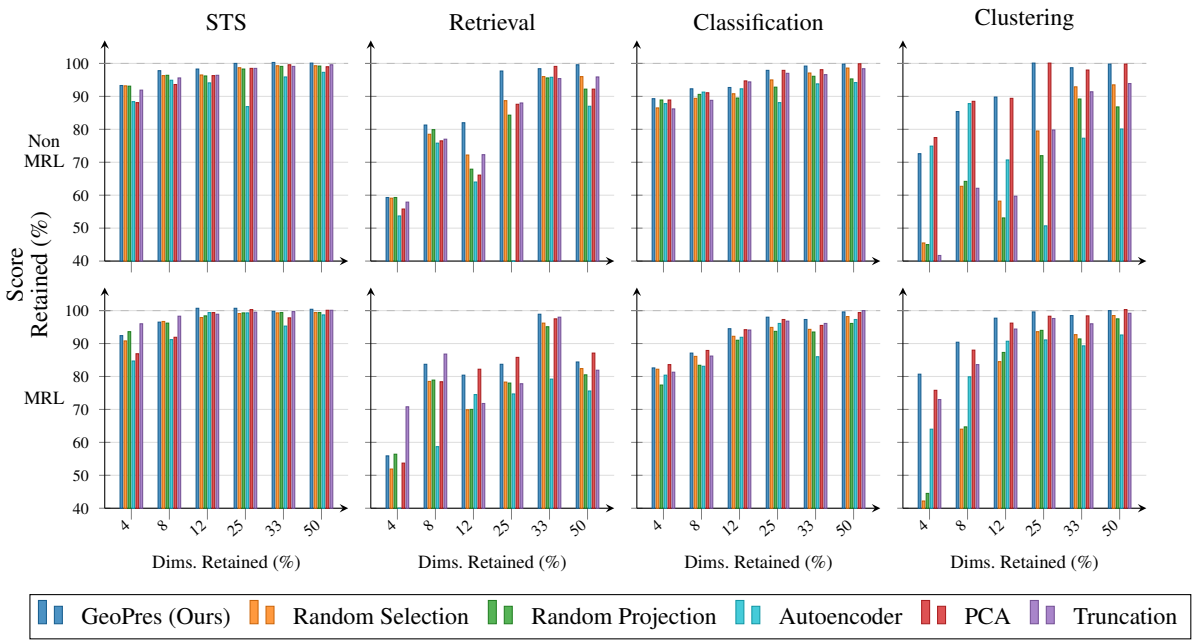


Figure 3: Fine-grained task performance (% retention, y-axis) of dimensionality reduction methods across degrees of reduction (x-axes). Top/bottom row: average of models without/with an MRL-trained backbone.

tion across all backbones. Second, the plot reveals the sensitivity of non-MRL backbones (Jina and MPNet) to truncation. For these models, truncation scores often fall below those of Random Selection or Random Projection at low ratios, confirming that without hierarchical training, the leading dimensions do not necessarily contain the most semantic information. Conversely, for MRL-enabled backbones (mGTE and Qwen), truncation remains highly effective, yet GEOPRES still manages to extract marginal but consistent gains, demonstrating that even in hierarchically structured spaces, GEOPRES can outperform coordinate slicing.

Individual Tasks and Matryoshka Backbones. Figure 3 gives a fine-grained overview of individual tasks, split by models with and without MRL-trained backbone. For most dimensionalities, GEOPRES outperforms all other methods. This also holds for MRL-trained backbones, suggesting MRL may not always be the best strategy, despite its popularity. We observe strong robustness for MRL, particularly at lower dimensions in retrieval. For all other tasks, GEOPRES tends to perform best on average.

6 Analysis

Intrinsic Metrics and Downstream Performance. Differences between reduction methods raise a geometric question: *Can we predict downstream task performance from how points are arranged in the reduced space?*

Answering this question could provide valuable insights into the relationship between geometric fidelity and downstream task utility. If certain spatial properties strongly correlate with extrinsic performance on MTEB tasks, we could use this information as a fast proxy for costly extrinsic benchmarking, and, more importantly, validate whether our design choices for the loss function are well-aligned with downstream task requirements.

To investigate this relationship, we compute the Spearman correlation coefficient between the average MTEB score and three intrinsic preservation metrics across all dimensionality reduction methods and target dimensions, computed on the held-out test set described in Section 4.2. The metrics are *Euclidean distance preservation* (Positional), *cosine similarity preservation* (Angular), and *cosine similarity rank preservation*[4] (Spearman), as described in Sections 3.3 and A.1.

The results are shown in Table 1, which reveals a striking result: Spearman Rank Loss on pairwise cosine similarities exhibits an exceptionally strong correlation with average MTEB performance ($\rho = 0.9360$, $p < 10^{-55}$). This correlation

[4]We adopt a *local* variant for computational tractability, computing the per-row rank correlation and averaging across all points.

Intrinsic Preservation Metric	Spearman ρ	p-value
Spearman	0.9360	$< 10^{-55}$
Positional	0.4954	$< 10^{-9}$
Angular	0.6292	$< 10^{-14}$

Table 1: Spearman correlation between intrinsic metrics and average MTEB score. All correlations are statistically significant ($p < 10^{-9}$).

is both statistically significant and practically meaningful. Note that ρ measures monotonic association (not variance explained); as a rough heuristic, $\rho^2 \approx 0.88$ for $\rho = 0.9360$.

Based on this insight, we tried directly optimizing a differentiable Spearman rank objective computed over the cosine-similarity matrices with the PyTorch sort package (Blondel et al., 2020). However, it shows slightly inferior MTEB results compared to our main method based on Euclidean distance preservation (Section A.1). We suspect this gap is partly attributable to optimization details rather than a contradiction of the correlation result: explicit hyperparameter tuning tailored to the Spearman-rank loss may yield higher performance.

However, given the strong correlation between Spearman Rank loss and MTEB performance, intrinsic evaluation using Spearman Rank loss could serve as a computationally efficient proxy for extrinsic benchmarking during method development. Computing Spearman Rank loss on a small validation set requires only the precomputed embeddings and takes seconds, whereas full MTEB evaluation requires running multiple downstream tasks and can take hours or days depending on the backbone model. Hence, we suggest that observing the internal rankings of embedding similarities may lead to good model development, but it should not be directly optimized without further exploration.

Cross-lingual Generalization. Our proposed method is general and makes few assumptions about the training data, working directly on the abstract representation space. However, since we trained on representations generated from monolingual English data, one might wonder if the reduction generalizes to other languages. For this experiment, we use the AmazonReviews classification dataset. This dataset is ideal for the purpose because it is parallel, and hence language-specific performances are immediately comparable.

For each non-English language and target dimension, we define retention as follows. First, we compute the accuracy of the original backbone model in the given language. Second, we compute the accuracy of the reduced model in the same language and dimension. The retention ratio $R^{\text{lang}}_{\text{dim}}$ is the reduced-model accuracy divided by the original-backbone accuracy. We then compute the retention delta relative to English: $\Delta^{\text{lang}}_{\text{dim}} = R^{\text{lang}}_{\text{dim}} - R^{\text{EN}}_{\text{dim}}$. Positive values indicate that performance is preserved better for the non–English language than for English.

Backbone	Dim	de	es	fr	ja	zh
mGTE	2	+0.085	+0.048	+0.104	+0.142	+0.221
	32	+0.061	+0.058	+0.065	+0.071	+0.110
	64	+0.007	+0.013	+0.019	+0.010	+0.016
	128	+0.031	+0.018	+0.023	+0.019	+0.017
	256	+0.013	+0.011	+0.009	+0.012	+0.010
Qwen	2	+0.062	+0.077	+0.098	+0.080	+0.239
	32	+0.024	+0.048	+0.045	+0.018	+0.074
	64	+0.044	+0.048	+0.053	+0.038	+0.065
	128	+0.008	+0.006	+0.016	+0.003	+0.028
	256	-0.007	-0.001	+0.005	+0.002	+0.004
	512	-0.003	+0.001	-0.000	-0.004	-0.002

Table 2: Retention $\Delta^{\text{lang}}_{\text{dim}}$ on AmazonReviews classification. Positive values suggest better performance preservation than English.

Table 2 shows the retention delta $\Delta^{\text{lang}}_{\text{dim}}$ (non-English minus English) for both multilingual backbones. We observe that non–English languages are not disadvantaged: at low dimensions, performance tends to be preserved slightly better than English, and the gap narrows to near zero as dimensionality increases. This pattern holds across both backbones and all five non–English languages.

We conclude that, on AmazonReviews classification, there is no evidence that English-only training harms retention for the tested non-English languages.

Extreme Case: Two Dimensions. Two-dimensional embeddings are an extreme case of dimensionality reduction, where substantial performance loss seems inevitable. Nevertheless, such extreme reductions are often used to visualize text embeddings. We therefore examine how the different methods behave under this extreme reduction setting. Table 3 reports the normalized MTEB scores at two dimensions across all four backbones and methods.

Several observations emerge. First, performance collapses dramatically across all methods: even the strongest configurations retain only around 35–38% of original MTEB score, confirming that 2D is an exceptionally harsh reduction. Second, classi-

	Method	MTEB	STS	Retr.	Class.	Clust.
mGTE	GeoPres	0.343	0.342	0.000	0.701	**0.293**
	Rand. Sel.	0.327	0.392	0.000	**0.705**	0.132
	Rand. Proj.	0.333	**0.453**	0.000	0.680	0.098
	Autoencoder	–	–	–	–	–
	PCA	**0.355**	0.394	0.000	0.703	0.272
	Truncation	0.348	0.441	0.000	0.699	0.169
Jina	GeoPres	0.367	0.364	**0.001**	0.751	**0.299**
	Rand. Sel.	0.345	0.385	0.000	0.766	0.123
	Rand. Proj.	0.358	0.396	**0.001**	**0.794**	0.135
	Autoencoder	–	–	–	–	–
	PCA	**0.379**	**0.417**	0.000	0.746	0.277
	Truncation	0.334	0.361	0.000	0.745	0.135
Qwen	GeoPres	0.364	0.348	0.000	**0.733**	**0.348**
	Rand. Sel.	0.336	0.406	0.000	0.717	0.141
	Rand. Proj.	0.341	0.416	0.000	0.728	0.137
	Autoencoder	0.330	0.425	0.000	0.654	0.161
	PCA	**0.374**	0.407	**0.001**	0.731	0.311
	Truncation	0.336	**0.430**	0.000	0.665	0.171
MPNet	GeoPres	**0.379**	0.393	0.001	0.811	**0.263**
	Rand. Sel.	0.366	**0.461**	0.001	0.776	0.130
	Rand. Proj.	0.360	0.442	0.001	0.787	0.119
	Autoencoder	0.342	0.335	0.001	0.788	0.202
	PCA	0.376	0.386	0.001	**0.819**	0.249
	Truncation	0.359	0.400	0.001	0.810	0.154

Table 3: Normalized MTEB scores at 2 dimensions across all backbones and methods. Autoencoder failed to converge for the mGTE and Jina backbones.

fication performance is surprisingly well retained, ranging from roughly 65% to 82% of the original scores, while retrieval scores are effectively zero across all methods. Third, GeoPres and, mainly, PCA remain the most robust methods at this extreme, consistently occupying the top positions across backbones. Fourth, truncation benefits from MRL backbones, but degrades on non-MRL ones, even though they were not trained to perform for 2 dimensions. Finally, the autoencoder fails to converge for Jina and mGTE entirely, and underperforms if it does converge, reinforcing our recommendation against reconstruction-based compression.

7 Discussion and Conclusions

Practical Recommendations for Embedding Compression. Our results suggest several practical guidelines for selecting dimensionality reduction methods for modern embedding systems.

1. Across most settings and tasks, GeoPres reduces dimensionality by 75% with *minimal* performance loss; for some tasks, such as STS and Clustering, score retention of up to 100% can be achieved.

2. GeoPres is a simple, linear, post-hoc geometry-preserving method that performs consistently across embedding backbones and target dimensions and outperforms all baselines on average. It is especially useful for models without native MRL support and for deployment scenarios requiring custom target dimensionalities, where coordinate-slicing methods like truncation are unreliable. To ease adoption, we provide it as a SentenceTransformer-compatible wrapper.

3. Truncation is robust, but only when the embedding model is explicitly trained with MRL and the target dimensionality matches a dimension supported during MRL training. If not, truncation often leads to substantially weaker retention.

4. Random selection and random projection are simple and computationally cheap, but consistently underperform all other approaches; random projection is more robust at very low dimensions, random selection more competitive at moderate and high ones. Autoencoder-based compression, under the same training setup as GeoPres, is harder to optimize and yields weaker retention, so we do not recommend it. PCA is a strong non-neural, linear baseline, especially at very low dimensions and for clustering, but it is less consistent than GeoPres across tasks and backbones.

Geometric Structure and Embedding Utility. Beyond the empirical comparison of reduction techniques, our results offer two broader insights into embedding compression.

1. Preserving cosine similarity *rankings* correlates with downstream task performance substantially more strongly than preserving Euclidean distances or angles. Rank-based intrinsic metrics can therefore serve as useful proxies for downstream evaluation when developing compression methods, providing meaningful estimates of quality at a fraction of the cost of full MTEB evaluation.

2. 2D reductions, though common for visualizing documents and clusters, should be interpreted with caution: our results show they cannot be achieved without severe loss of representational information.

Limitations

Although each individual training run requires only a few minutes on consumer hardware, our experiments required substantial computational resources because we varied multiple factors, including embedding model, dimensionality reduction technique, target dimensionality, and NLP task. This large-scale experimentation allowed us to derive general conclusions and recommendations. Nevertheless, our study is not exhaustive, and certain target dimensionalities, backbone models, or task settings may exhibit different patterns.

Our projection models are trained on English C4 data. Although our cross-lingual experiment suggests that this does not harm retention on AmazonReviews classification for the tested multilingual backbones, this result may not generalize to all languages, domains, or tasks. Specialized corpora, such as historical, biomedical, or legal text, may induce different embedding-space geometry.

We also evaluate only four embedding backbones and a selected set of MTEB task groups. These cover common embedding use cases, but they do not cover all embedding systems or deployment scenarios, such as sparse or hybrid retrievers, long-document retrieval, reranking pipelines, or production-scale retrieval under strict latency and memory constraints.

References

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

Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. 2020. Fast differentiable sorting and ranking. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 950–959. PMLR.

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 (EMNLP)*, pages 6894–6910. Association for Computational Linguistics.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-Token general-purpose text embeddings for long documents. *Preprint*, arXiv:2310.19923.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. 24(6):417–441.

William B. Johnson and Joram Lindenstrauss. 1984. Extensions of lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *Preprint*, arXiv:2202.08904.

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, Dubrovnik, Croatia. Association for Computational Linguistics.

Juri Opitz, Lucas Moeller, Andrianos Michail, Sebastian Padó, and Simon Clematide. 2025. Interpretable text embeddings and text similarity explanation: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22303–22319, Suzhou, China. Association for Computational Linguistics.

Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

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.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Sotaro Takeshita, Yurina Takeshita, Daniel Ruffinelli, and Simone Paolo Ponzetto. 2025. Randomly removing 50% of dimensions in text embeddings has minimal impact on retrieval and classification tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27705–27726, Suzhou, China. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Roman Vershynin. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US. Association for Computational Linguistics.

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

A Appendix

A.1 Alternative Loss Functions

We evaluate two alternative loss functions to validate our choice of Euclidean distance preservation as the primary training objective. Comprehensive experiments were conducted using the `GTE-Multilingual-Base` and `Jina-Embeddings-V2-Small-En` backbones, which span both MRL-enabled and non-MRL configurations. Due to computational constraints, we did not extend these experiments to the remaining backbones. After identifying the best-performing configuration through this ablation study (Section 3.3), we adopted it as the primary method throughout this work.

A.1.1 Cosine Similarity Preservation

As an alternative to preserving Euclidean distances, we experimented with directly preserving pairwise cosine similarities. Given a batch $B = \{y_1, y_2, \ldots, y_m\} \subseteq \mathbb{R}^n$, we first ℓ_2-normalize both the high- and low-dimensional embeddings, and compute the corresponding similarity matrices S and S^{f}, where $s_{i,j} = \langle \tilde{y}_i, \tilde{y}_j \rangle$ and $\tilde{y}$ denotes the normalized vector. The loss is then defined as the mean squared error over the upper-triangular entries:

$$\mathcal{L}_{\cos} = \frac{1}{\binom{m}{2}} \sum_{i<j} \left(s_{i,j} - s^{\mathrm{f}}_{i,j} \right)^2 . \qquad (3)$$

A.1.2 Spearman Rank Preservation

We also explored optimizing directly for rank-order preservation of pairwise similarities, rather than their absolute values. Using the differentiable soft ranking operator of Blondel et al. (2020), we compute a smooth approximation of Spearman's ρ between the upper-triangular entries of S and S^{f}. [5] The loss is defined as $\mathcal{L}_{\mathrm{spr}} = 1 - \rho$, where ρ is the differentiable Spearman correlation.

A.1.3 Comparison against Main Method

Similarly to Section 4, we evaluated on MTEB, reporting (now un-normalized) scores across STS, Retrieval, Classification, and Clustering tasks.

Tables 4 and 5 present the MTEB evaluation results comparing all alternative loss function variants against the loss function used in our main method (positional loss). The positional loss achieves the highest MTEB performance across most configurations, confirming our selection of this variant as our main method.

Overall, the extrinsic evaluation results confirm that the distance-preserving (positional) loss is the most effective training objective among all variants tested. While optimizing for cosine similarity or rank-order preservation seems like the desired objective intuitively, the results disagree.

[5] We adopt a *local* variant as described in Section 6.

Method	MTEB	AVG STS	AVG Retr.	AVG Class.	AVG Clust.
Dim = 32					
Angular	0.517	0.749	0.343	0.547	0.431
Spearman	0.511	0.749	0.325	0.553	0.418
Positional	**0.527**	**0.771**	**0.349**	**0.555**	**0.433**
Dim = 64					
Angular	0.592	0.794	0.508	0.581	**0.485**
Spearman	0.580	0.787	0.479	0.579	0.476
Positional	**0.600**	**0.805**	**0.523**	**0.585**	**0.485**
Dim = 128					
Angular	0.634	0.821	0.592	0.608	**0.516**
Spearman	0.625	0.813	0.576	0.601	0.510
Positional	**0.637**	**0.823**	**0.595**	**0.618**	0.512
Dim = 256					
Angular	**0.658**	**0.834**	**0.621**	0.647	0.529
Spearman	0.655	0.831	0.618	0.637	**0.532**
Positional	**0.658**	0.833	0.618	**0.654**	0.528

Table 4: Extrinsic evaluation: alternative loss functions for `GTE-Multilingual-Base`. Higher is better.

Method	MTEB	AVG STS	AVG Retr.	AVG Class.	AVG Clust.
Dim = 32					
Angular	**0.500**	0.758	**0.310**	0.566	**0.366**
Spearman	0.495	0.754	0.304	**0.572**	0.350
Positional	0.498	**0.761**	**0.310**	0.561	0.359
Dim = 64					
Angular	**0.566**	0.794	0.464	**0.593**	0.414
Spearman	0.565	0.789	0.462	0.585	**0.422**
Positional	0.562	**0.797**	**0.465**	0.581	0.407
Dim = 128					
Angular	0.599	0.808	0.536	**0.614**	0.437
Spearman	**0.608**	0.809	0.552	0.612	**0.458**
Positional	**0.608**	**0.811**	**0.554**	0.613	0.454
Dim = 256					
Angular	**0.614**	**0.812**	0.565	**0.626**	**0.453**
Spearman	**0.614**	**0.812**	**0.566**	**0.626**	0.451
Positional	**0.614**	**0.812**	0.565	0.625	**0.453**

Table 5: Extrinsic evaluation: alternative loss functions for `Jina-Embeddings-V2-Small-En`. Higher is better.

A.1.4 Raw MTEB Scores

Previously, we reported normalized, average MTEB results (see Section 4.4). For completeness and to enable direct comparison with other work that reports the original metrics, we include the corresponding non-normalized MTEB results below. Here, RP denotes Random Projection, RS Random Selection, AE Autoencoder, and GeoPres our proposed method.

Dim	Method	STS12	STS13	STS14	STS15	STS16	STSBenchmark	SICK-R	QuoraRetrieval	HotpotQA	DBPedia	NQ	MSMARCO	ArguAna	AmazonCounterfactualClassification	AmazonPolarityClassification	AmazonReviewsClassification	ImdbClassification	ToxicConversationsClassification	ArxivClusteringS2S	RedditClustering	StackExchangeClustering
768	–	0.7751	0.8555	0.8168	0.8894	0.8434	0.8646	0.7934	0.8801	0.6303	0.4011	0.5809	0.6751	0.5834	0.7155	0.8057	0.3993	0.7568	0.6831	0.4105	0.5577	0.6413
256	PCA	0.752	0.824	0.788	0.873	0.821	0.844	**0.807**	0.875	0.610	0.393	0.569	0.651	0.562	0.652	0.780	0.389	0.720	0.669	**0.405**	0.551	0.627
	RP	0.769	0.845	0.808	0.881	**0.845**	0.862	0.791	0.874	0.597	0.365	0.544	0.641	0.546	0.679	0.762	0.364	0.694	0.644	0.389	0.505	0.577
	RS	0.772	0.855	0.812	0.881	0.836	0.857	0.787	**0.878**	0.605	0.371	0.559	0.640	0.557	0.680	0.750	0.373	0.705	0.660	0.395	0.513	0.583
	Trunc	**0.773**	**0.857**	**0.814**	0.884	0.841	0.863	0.791	0.877	0.613	0.378	0.566	**0.666**	0.576	**0.684**	0.772	0.385	0.719	0.669	0.400	0.527	0.617
	AE	0.722	0.815	0.758	0.840	0.814	0.826	0.786	0.841	0.398	0.227	0.446	0.510	0.550	0.593	0.695	0.357	0.643	0.603	0.368	0.499	0.570
	GeoPres	**0.773**	0.852	0.812	**0.887**	0.842	**0.865**	0.796	0.877	**0.620**	**0.395**	**0.574**	0.659	**0.584**	0.677	**0.787**	**0.392**	**0.731**	**0.682**	0.403	**0.553**	**0.630**
128	PCA	0.729	0.801	0.759	0.852	0.803	0.821	**0.803**	0.862	0.551	0.347	0.536	0.603	0.541	0.609	**0.742**	**0.375**	0.682	0.631	**0.395**	0.527	0.605
	RP	0.761	0.834	0.804	0.872	0.834	0.854	0.778	0.866	0.559	0.318	0.510	0.600	0.523	0.653	0.714	0.346	0.658	0.634	0.368	0.439	0.515
	RS	**0.774**	0.849	**0.805**	0.871	0.824	0.845	0.786	0.869	0.559	0.337	0.512	0.607	0.533	**0.661**	0.717	0.345	0.682	0.634	0.368	0.454	0.499
	Trunc	0.768	**0.852**	**0.805**	**0.881**	**0.836**	**0.858**	0.789	**0.873**	**0.583**	0.344	0.538	0.640	0.551	0.648	0.737	0.373	**0.692**	0.637	0.388	0.508	0.581
	AE	0.735	0.821	0.762	0.847	0.819	0.833	0.790	0.844	0.423	0.247	0.462	0.533	**0.562**	0.592	0.690	0.356	0.639	0.614	0.374	0.504	0.580
	GeoPres	0.759	0.842	0.802	0.875	0.835	0.855	0.795	0.871	**0.583**	**0.357**	**0.550**	**0.644**	**0.562**	0.644	0.736	0.372	0.679	**0.656**	0.393	**0.538**	**0.606**
64	PCA	0.697	0.771	0.724	0.826	0.774	0.789	**0.786**	0.837	0.399	0.238	0.449	0.508	0.511	0.593	**0.717**	**0.365**	**0.665**	0.615	**0.368**	0.480	0.569
	RP	0.751	0.819	0.785	0.855	0.811	0.834	0.764	0.851	0.460	0.248	0.426	0.517	0.459	0.624	0.645	0.314	0.614	0.606	0.324	0.323	0.395
	RS	**0.765**	0.820	0.781	0.856	0.810	0.831	0.778	0.853	0.454	0.244	0.417	0.511	0.465	**0.641**	0.678	0.320	0.626	**0.628**	0.318	0.341	0.370
	Trunc	**0.765**	**0.846**	**0.794**	**0.867**	**0.833**	**0.856**	0.780	**0.861**	**0.506**	**0.283**	**0.481**	**0.603**	0.521	0.604	0.683	0.350	0.641	0.620	0.363	0.457	0.526
	AE	0.686	0.767	0.710	0.821	0.788	0.792	0.760	0.793	0.202	0.124	0.279	0.336	0.469	0.590	0.659	0.342	0.614	0.586	0.331	0.438	0.517
	GeoPres	0.742	0.823	0.783	0.854	0.820	0.837	0.774	0.853	0.464	0.262	0.475	0.552	**0.532**	0.605	0.704	0.358	0.642	0.620	**0.368**	**0.508**	**0.580**
32	PCA	0.644	0.725	0.674	0.796	0.742	0.741	0.752	0.780	0.180	0.109	0.258	0.278	0.408	0.585	**0.675**	**0.348**	**0.621**	0.582	0.320	0.404	0.497
	RP	0.733	0.804	0.761	0.825	0.778	0.819	0.746	0.800	0.241	0.111	0.237	0.373	0.352	0.573	0.589	0.288	0.578	0.572	0.246	0.217	0.253
	RS	0.729	0.755	0.724	0.821	0.757	0.779	0.736	0.790	0.210	0.098	0.227	0.314	0.306	**0.598**	0.651	0.313	0.616	0.586	0.245	0.216	0.218
	Trunc	**0.748**	**0.822**	**0.769**	**0.845**	**0.817**	**0.836**	**0.769**	**0.832**	**0.327**	**0.176**	**0.358**	**0.507**	**0.457**	0.562	0.634	0.322	0.617	**0.598**	0.324	0.395	0.456
	AE	0.619	0.706	0.649	0.780	0.732	0.726	0.730	0.667	0.056	0.034	0.097	0.126	0.303	0.572	0.630	0.318	0.617	0.567	0.261	0.342	0.428
	GeoPres	0.706	0.794	0.738	0.822	0.797	0.806	0.735	0.803	0.199	0.094	0.266	0.299	0.435	0.589	0.649	0.329	0.616	0.592	**0.328**	**0.441**	**0.529**
2	PCA	0.284	0.338	0.270	0.348	0.351	0.344	0.363	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	**0.540**	0.531	0.240	0.534	0.518	**0.132**	0.128	0.178
	RP	0.359	**0.419**	**0.337**	0.384	0.362	**0.411**	0.373	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.000	0.523	0.529	0.230	0.536	0.466	0.081	0.031	0.046
	RS	0.312	0.262	0.284	**0.387**	0.372	0.292	**0.381**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	0.511	**0.570**	0.246	0.513	0.528	0.095	0.061	0.057
	Trunc	**0.452**	0.322	0.290	0.381	**0.381**	0.379	0.368	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	0.505	0.523	0.242	0.531	**0.547**	0.130	0.069	0.073
	AE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	GeoPres	0.198	0.322	0.216	0.264	0.369	0.271	0.356	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.001**	0.498	0.564	**0.247**	**0.558**	0.489	0.124	**0.144**	**0.203**

Table 6: Raw task scores for `Alibaba-NLP/gte-multilingual-base`. The autoencoder failed to converge at a target dimension of 2.

Dim	Method	STS12	STS13	STS14	STS15	STS16	STSBenchmark	SICK-R	QuoraRetrieval	HotpotQA	DBPedia	NQ	MSMARCO	ArguAna	AmazonCounterfactualClassification	AmazonPolarityClassification	AmazonReviewsClassification	ImdbClassification	ToxicConversationsClassification	ArxivClusteringS2S	RedditClustering	StackExchangeClustering
512	–	0.7366	0.8330	0.7917	0.8730	0.8360	0.8404	0.7672	0.8718	0.5648	0.3265	0.5156	0.6533	0.4674	0.6598	0.8242	0.2778	0.7087	0.6596	0.3456	0.4757	0.5399
256	PCA	0.722	0.818	0.783	0.865	0.831	0.832	**0.768**	0.870	0.502	0.276	0.428	0.628	0.431	**0.657**	**0.824**	0.277	0.707	**0.660**	0.342	**0.474**	0.543
	RP	0.733	0.826	0.787	0.864	0.827	0.831	0.766	0.864	0.477	0.274	0.455	0.622	0.441	0.625	0.795	0.269	0.676	0.617	0.303	0.409	0.469
	RS	0.736	0.822	0.785	0.864	0.836	0.835	0.762	0.869	0.520	0.294	0.489	0.637	0.453	0.649	0.807	0.274	0.700	0.655	0.328	0.441	0.504
	Trunc	0.739	0.829	0.786	0.869	0.833	0.837	0.764	0.867	0.522	0.296	0.493	0.635	0.447	0.655	0.811	0.276	0.694	0.643	0.325	0.446	0.507
	AE	**0.747**	0.812	0.766	0.844	0.816	0.808	0.733	0.842	0.405	0.281	0.416	0.573	0.440	0.594	0.786	0.262	0.673	0.632	0.257	0.398	0.435
	GeoPres	0.738	**0.834**	**0.793**	**0.873**	**0.837**	**0.841**	**0.768**	**0.871**	**0.563**	**0.321**	**0.515**	**0.649**	**0.468**	0.656	0.823	**0.278**	**0.709**	0.658	**0.345**	0.470	**0.544**
128	PCA	0.730	0.816	0.781	0.861	0.823	0.827	0.756	0.856	0.455	0.266	0.379	0.603	0.418	0.628	**0.817**	0.273	0.697	0.649	0.332	0.477	**0.553**
	RP	0.733	0.816	0.781	0.851	0.813	0.826	**0.763**	0.855	0.388	0.214	0.396	0.598	0.414	0.608	0.774	0.262	0.661	0.599	0.259	0.342	0.380
	RS	0.740	0.816	0.779	0.854	0.827	0.828	0.758	0.860	0.430	0.243	0.430	0.621	0.430	0.628	0.764	0.267	0.664	0.649	0.285	0.378	0.419
	Trunc	0.730	0.809	0.778	0.860	0.827	0.830	0.760	0.858	0.427	0.250	0.442	0.606	0.408	**0.633**	0.797	0.270	0.682	**0.655**	0.277	0.377	0.431
	AE	0.652	0.713	0.667	0.752	0.732	0.726	0.694	0.612	0.016	0.010	0.027	0.047	0.161	0.560	0.728	0.241	0.641	0.587	0.148	0.252	0.289
	GeoPres	**0.743**	**0.838**	**0.793**	**0.867**	**0.835**	**0.837**	0.762	**0.861**	**0.528**	**0.325**	**0.490**	**0.653**	**0.465**	0.631	**0.817**	**0.274**	**0.701**	0.641	**0.335**	**0.480**	0.547
64	PCA	0.717	0.798	0.757	0.838	0.801	0.805	0.751	0.835	0.243	0.133	0.231	0.442	0.363	0.577	**0.813**	**0.264**	**0.699**	0.609	0.262	**0.455**	**0.501**
	RP	0.715	0.789	0.763	0.835	0.798	0.807	**0.756**	0.831	0.244	0.130	0.288	0.475	0.341	0.588	0.759	0.255	0.636	0.562	0.205	0.244	0.274
	RS	0.730	0.800	0.756	0.832	0.808	0.810	0.743	0.838	0.266	0.153	0.299	0.538	0.359	**0.619**	0.714	0.254	0.635	0.620	0.213	0.264	0.315
	Trunc	0.725	0.791	0.764	0.835	0.804	0.809	0.746	0.838	0.251	0.165	0.317	0.547	0.342	0.618	0.752	0.260	0.668	**0.656**	0.220	0.277	0.317
	AE	0.719	0.783	0.736	0.816	0.790	0.780	0.720	0.807	0.177	0.141	0.253	0.444	0.355	0.563	0.778	0.252	0.675	0.622	0.199	0.364	0.399
	GeoPres	**0.734**	**0.822**	**0.777**	**0.850**	**0.821**	**0.820**	**0.756**	**0.843**	**0.340**	**0.230**	**0.384**	**0.569**	**0.421**	0.576	0.791	0.262	0.679	0.595	**0.275**	0.448	0.500
32	PCA	0.680	0.752	0.711	0.794	0.765	0.767	**0.741**	0.775	0.071	0.024	0.077	0.174	0.256	0.566	**0.807**	**0.256**	**0.708**	0.593	0.194	0.369	0.409
	RP	0.688	0.764	0.724	0.800	0.762	0.764	0.734	0.757	0.082	0.038	0.123	0.274	0.240	0.552	0.702	0.237	0.622	0.551	0.162	0.159	0.190
	RS	**0.713**	0.763	0.735	0.799	0.779	0.777	0.730	0.771	0.077	0.042	0.119	0.337	0.245	0.588	0.674	0.246	0.633	0.609	0.160	0.161	0.198
	Trunc	0.699	0.749	**0.736**	0.796	0.773	0.766	0.727	0.764	0.078	0.052	0.124	0.298	0.235	**0.593**	0.705	0.244	0.656	**0.641**	0.157	0.176	0.209
	AE	0.625	0.697	0.645	0.747	0.721	0.707	0.689	0.513	0.007	0.004	0.013	0.005	0.123	0.553	0.743	0.244	0.656	0.574	0.142	0.257	0.297
	GeoPres	0.711	**0.778**	0.731	**0.820**	**0.785**	**0.783**	0.722	**0.796**	**0.114**	**0.081**	**0.182**	**0.338**	**0.350**	0.566	0.751	0.246	0.659	0.585	**0.226**	**0.398**	**0.453**
2	PCA	**0.390**	0.338	0.272	0.288	0.363	**0.392**	0.323	**0.000**	**0.000**	0.000	**0.000**	**0.000**	0.000	0.512	0.562	**0.225**	0.518	0.519	0.106	0.116	0.155
	RP	0.267	0.295	**0.279**	0.339	**0.385**	0.337	**0.346**	**0.000**	**0.000**	0.000	**0.000**	**0.000**	**0.003**	0.506	**0.620**	0.217	**0.600**	0.540	0.084	0.038	0.061
	RS	0.359	0.288	0.277	0.243	0.369	0.309	0.338	**0.000**	**0.000**	0.000	**0.000**	**0.000**	0.001	0.524	0.575	0.222	0.561	0.515	0.081	0.038	0.048
	Trunc	0.357	0.275	0.234	0.317	0.324	0.257	0.285	**0.000**	**0.000**	0.000	**0.000**	**0.000**	0.001	0.505	0.537	0.210	0.511	**0.567**	0.087	0.046	0.051
	AE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	GeoPres	0.218	**0.373**	0.256	**0.340**	0.354	0.234	0.294	**0.000**	**0.000**	**0.001**	**0.000**	**0.000**	0.002	**0.528**	0.533	0.216	0.544	0.531	**0.112**	**0.132**	**0.164**

Table 7: Raw task scores for `jinaai/jina-embeddings-v2-small-en`. The autoencoder failed to converge at a target dimension of 2.

Dim	Method	STS12	STS13	STS14	STS15	STS16	STSBenchmark	SICK-R	QuoraRetrieval	HotpotQA	DBPedia	NQ	MSMARCO	ArguAna	AmazonCounterfactualClassification	AmazonPolarityClassification	AmazonReviewsClassification	ImdbClassification	ToxicConversationsClassification	ArxivClusteringS2S	RedditClustering	StackExchangeClustering
1024	–	0.7712	0.8006	0.7677	0.8607	0.8410	0.8460	0.8116	0.8718	0.6515	0.3925	0.5303	0.6699	0.6755	0.7481	0.8238	0.3863	0.7994	0.6436	0.4388	0.5123	0.6443
256	PCA	0.767	**0.821**	0.768	0.860	0.844	0.847	0.809	0.873	0.547	**0.225**	**0.399**	**0.538**	0.671	0.703	0.810	0.379	0.785	0.632	0.429	0.509	0.631
	RP	0.763	0.801	0.759	0.850	0.842	0.842	0.803	0.869	0.509	0.157	0.311	0.462	0.649	0.710	0.788	0.353	0.734	0.603	0.425	0.469	0.606
	RS	0.759	0.797	0.760	0.851	0.834	0.836	0.807	0.872	0.512	0.142	0.311	0.476	0.655	**0.713**	0.768	0.365	0.752	0.632	0.427	0.468	0.597
	Trunc	0.767	0.799	0.756	0.856	0.836	0.842	**0.812**	0.868	0.501	0.150	0.304	0.463	0.664	0.689	**0.818**	**0.385**	0.783	0.619	0.432	0.489	0.635
	AE	**0.793**	0.811	0.764	0.831	0.837	0.826	0.796	0.866	0.492	0.109	0.286	0.407	**0.672**	0.709	0.781	0.369	**0.789**	0.621	0.419	0.450	0.583
	GeoPres	0.779	0.819	**0.775**	**0.861**	**0.848**	**0.849**	0.811	**0.875**	**0.567**	0.179	0.366	0.514	**0.672**	**0.713**	0.815	0.381	0.787	**0.637**	**0.433**	**0.516**	**0.640**
128	PCA	0.760	0.824	0.758	0.851	0.833	0.838	0.800	0.864	0.456	**0.227**	**0.376**	**0.537**	0.655	0.670	0.787	0.368	0.757	0.621	0.414	0.503	0.618
	RP	0.750	0.799	0.751	0.841	0.834	0.837	0.795	0.862	0.405	0.126	0.261	0.394	0.607	0.686	0.758	0.338	0.713	0.601	0.415	0.425	0.553
	RS	0.754	0.788	0.746	0.839	0.828	0.827	0.797	0.863	0.397	0.100	0.244	0.431	0.616	**0.699**	0.740	0.351	0.720	**0.628**	0.411	0.413	0.523
	Trunc	0.765	0.794	0.750	0.849	0.830	0.840	**0.810**	0.861	0.420	0.131	0.244	0.419	0.644	0.660	**0.795**	**0.384**	**0.764**	0.597	**0.426**	0.462	0.618
	AE	**0.798**	0.824	0.757	0.834	0.832	0.824	0.794	0.860	0.421	0.138	0.302	0.457	0.646	0.660	0.763	0.360	0.742	0.599	0.400	0.466	0.582
	GeoPres	0.779	**0.830**	**0.773**	**0.859**	**0.844**	**0.848**	0.805	**0.868**	**0.482**	0.178	0.362	0.502	**0.656**	0.673	0.792	0.372	0.754	0.622	0.423	**0.505**	**0.630**
64	PCA	0.746	0.807	0.738	0.832	0.817	0.823	0.792	0.846	0.295	**0.168**	**0.288**	**0.463**	0.605	0.625	**0.771**	0.359	0.722	0.586	0.378	0.472	0.576
	RP	0.755	0.781	0.723	0.823	0.819	0.817	0.782	0.844	0.230	0.073	0.160	0.296	0.520	0.655	0.737	0.323	0.670	0.555	0.389	0.324	0.451
	RS	0.742	0.766	0.726	0.815	0.811	0.811	0.777	0.841	0.205	0.050	0.127	0.311	0.532	**0.660**	0.705	0.318	0.688	0.554	0.378	0.319	0.397
	Trunc	0.746	0.794	0.737	0.834	0.819	0.827	**0.803**	0.845	0.286	0.096	0.188	0.306	0.595	0.617	0.770	**0.362**	**0.736**	0.594	**0.407**	0.423	0.578
	AE	**0.773**	0.806	0.738	0.827	0.815	0.827	0.787	0.842	0.242	0.124	0.230	0.426	0.599	0.618	0.756	0.356	0.714	0.589	0.373	0.465	0.567
	GeoPres	0.765	**0.817**	**0.757**	**0.843**	**0.830**	**0.836**	0.801	**0.853**	**0.309**	0.143	0.281	0.435	**0.609**	0.631	0.763	0.354	0.690	**0.600**	0.392	**0.473**	**0.598**
32	PCA	0.704	0.783	0.703	0.805	0.793	0.791	0.780	0.802	0.117	**0.063**	**0.152**	**0.247**	**0.520**	0.603	0.754	**0.352**	0.682	0.581	0.339	0.424	0.537
	RP	0.716	0.752	0.684	0.789	0.781	0.781	0.756	0.784	0.072	0.019	0.060	0.102	0.388	0.612	0.658	0.294	0.661	0.579	0.338	0.224	0.301
	RS	0.711	0.743	0.694	0.778	0.786	0.779	0.745	0.763	0.056	0.013	0.031	0.115	0.350	**0.646**	0.709	0.294	0.673	0.558	0.307	0.208	0.269
	Trunc	**0.735**	0.776	0.710	**0.817**	0.794	0.803	**0.788**	**0.809**	**0.123**	0.056	0.116	0.226	0.498	0.577	0.747	0.351	**0.716**	0.583	**0.379**	0.382	0.526
	AE	0.732	0.761	0.689	0.799	0.787	0.790	0.776	0.781	0.068	0.036	0.082	0.126	0.478	0.593	**0.758**	0.346	0.691	0.565	0.300	0.411	0.500
	GeoPres	0.719	**0.784**	**0.720**	0.810	**0.802**	**0.809**	0.787	0.808	0.109	0.057	0.127	0.221	0.516	0.611	0.748	0.343	0.669	**0.586**	0.369	**0.426**	**0.568**
2	PCA	0.305	**0.356**	0.269	0.331	0.361	0.329	0.367	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.003**	0.535	0.568	**0.253**	0.606	0.523	0.148	0.150	0.199
	RP	0.400	0.285	**0.287**	0.350	0.329	0.340	0.378	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.561	0.573	0.237	0.563	**0.542**	0.107	0.049	0.062
	RS	0.386	0.271	0.267	0.330	0.326	0.352	0.378	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.001	**0.613**	0.567	0.236	0.543	0.479	0.114	0.045	0.065
	Trunc	**0.435**	0.292	0.273	0.329	0.361	0.382	**0.380**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.499	0.497	0.221	0.526	0.518	0.115	0.072	0.086
	AE	0.366	0.291	0.285	0.331	**0.385**	**0.387**	0.374	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.000	0.511	0.507	0.203	0.514	0.489	0.116	0.068	0.072
	GeoPres	0.225	0.205	0.236	**0.365**	0.330	0.289	0.337	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	0.000	0.543	**0.586**	0.239	**0.618**	0.506	**0.160**	**0.159**	**0.237**

Table 8: Raw task scores for `Qwen/Qwen3-Embedding-0.6B`.

Dim	Method	STS12	STS13	STS14	STS15	STS16	STSBenchmark	SICK-R	QuoraRetrieval	HotpotQA	DBPedia	NQ	MSMARCO	ArguAna	AmazonCounterfactualClassification	AmazonPolarityClassification	AmazonReviewsClassification	ImdbClassification	ToxicConversationsClassification	ArxivClusteringS2S	RedditClustering	StackExchangeClustering
768	–	0.7263	0.8348	0.7800	0.8566	0.8003	0.8342	0.8059	0.8746	0.3929	0.3209	0.5045	0.6668	0.4654	0.6222	0.6714	0.2683	0.7117	0.6105	0.3935	0.5463	0.5365
256	PCA	0.719	0.832	0.775	0.853	0.797	0.831	**0.811**	**0.874**	**0.380**	**0.327**	**0.500**	0.654	0.462	0.603	0.663	0.265	0.698	0.601	0.391	0.540	0.517
	RP	0.717	0.829	0.774	0.848	0.796	0.828	0.797	0.870	0.328	0.289	0.477	**0.675**	0.444	0.600	0.661	0.257	0.677	0.575	0.380	0.489	0.448
	RS	0.724	0.832	0.777	0.849	0.792	0.826	0.801	0.871	0.340	0.299	0.483	0.648	0.456	**0.613**	0.641	0.256	0.693	0.596	0.383	0.508	0.480
	Trunc	0.720	0.827	0.771	0.848	0.795	0.827	0.800	0.870	0.343	0.286	0.480	0.651	0.447	0.610	0.644	0.261	0.682	0.589	0.384	0.500	0.467
	AE	0.695	0.809	0.758	0.836	0.744	0.806	0.759	0.862	0.367	0.312	0.492	0.627	0.429	0.576	0.625	0.254	0.679	0.573	0.351	0.430	0.360
	GeoPres	**0.731**	**0.838**	**0.783**	**0.856**	**0.803**	**0.839**	0.808	**0.874**	0.375	0.313	0.497	0.652	**0.463**	0.610	**0.669**	**0.266**	**0.704**	**0.611**	**0.395**	**0.544**	**0.518**
128	PCA	0.704	0.816	0.751	0.833	0.771	0.808	0.803	0.864	0.290	**0.296**	0.464	0.599	0.451	0.571	0.634	0.255	0.667	0.569	0.376	0.525	**0.494**
	RP	0.710	0.826	0.769	0.840	0.791	0.823	0.789	0.864	0.275	0.253	0.445	**0.658**	0.420	0.588	**0.645**	0.254	0.648	0.559	0.360	0.424	0.376
	RS	0.718	0.821	0.766	0.839	0.784	0.817	0.792	0.865	0.266	0.244	0.452	0.623	0.443	0.602	0.606	0.247	0.645	0.575	0.360	0.440	0.401
	Trunc	0.712	0.819	0.758	0.836	0.783	0.817	0.794	0.865	0.265	0.249	0.449	0.619	0.435	**0.604**	0.621	0.254	0.651	**0.586**	0.361	0.438	0.396
	AE	0.722	0.825	0.756	0.834	0.791	0.829	0.792	0.863	0.272	0.272	0.455	0.603	0.441	0.561	0.615	0.253	0.656	0.566	0.356	0.506	0.443
	GeoPres	**0.730**	**0.833**	**0.776**	**0.846**	**0.796**	**0.834**	**0.806**	**0.868**	**0.307**	0.292	**0.475**	0.632	**0.456**	0.581	0.641	**0.261**	**0.682**	0.585	**0.385**	**0.527**	0.480
64	PCA	0.664	0.784	0.714	0.809	0.744	0.773	0.791	0.841	0.141	0.172	0.388	0.522	0.402	0.559	0.617	0.249	0.657	0.544	0.350	0.497	**0.460**
	RP	0.698	0.801	0.748	0.817	**0.790**	0.804	0.777	0.850	**0.177**	0.179	0.383	**0.615**	0.375	0.555	**0.645**	0.246	0.638	0.529	0.319	0.340	0.289
	RS	0.704	0.801	0.749	**0.825**	0.770	0.803	0.777	0.850	0.153	0.160	0.387	0.588	0.392	0.573	0.572	0.242	0.623	0.568	0.307	0.344	0.275
	Trunc	0.689	0.794	0.738	0.818	0.770	0.801	0.779	0.850	0.150	0.143	0.365	0.593	0.381	**0.575**	0.591	0.241	0.594	0.561	0.306	0.329	0.282
	AE	0.675	0.793	0.720	0.812	0.770	0.797	0.786	0.841	0.138	0.175	0.374	0.517	0.400	0.560	0.615	0.249	**0.658**	0.550	0.346	**0.501**	0.449
	GeoPres	**0.716**	**0.816**	**0.757**	**0.825**	0.781	**0.818**	**0.801**	**0.854**	0.172	**0.197**	**0.407**	0.569	**0.425**	0.571	0.606	**0.251**	**0.658**	**0.576**	**0.355**	0.490	0.415
32	PCA	0.612	0.742	0.658	0.777	0.706	0.719	0.756	0.792	0.039	0.054	0.235	0.356	0.323	0.543	0.609	0.239	**0.640**	0.532	0.312	**0.428**	**0.404**
	RP	0.674	**0.773**	0.715	0.780	**0.776**	0.781	0.749	0.806	**0.059**	0.069	0.246	0.424	0.310	0.556	**0.614**	0.233	0.630	0.531	0.255	0.227	0.183
	RS	0.680	0.769	**0.725**	**0.797**	0.749	0.779	0.759	0.802	0.055	0.063	0.247	0.417	0.322	0.556	0.570	0.229	0.597	0.542	0.251	0.241	0.179
	Trunc	0.663	0.743	0.703	0.794	0.747	0.777	0.753	0.801	0.052	0.043	0.226	**0.435**	0.309	**0.557**	0.574	0.233	0.571	0.551	0.231	0.208	0.177
	AE	0.614	0.739	0.649	0.773	0.725	0.732	0.753	0.789	0.033	0.052	0.221	0.340	0.296	0.539	0.589	0.236	0.635	0.534	0.299	0.419	0.388
	GeoPres	**0.690**	0.770	0.712	0.787	0.751	**0.783**	**0.770**	**0.809**	0.046	**0.073**	**0.256**	0.371	**0.358**	0.543	0.590	**0.245**	0.637	**0.560**	**0.315**	0.407	0.350
2	PCA	0.340	0.331	0.274	0.259	0.346	0.279	0.345	**0.001**	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.511	**0.573**	**0.211**	0.570	0.497	0.113	0.136	**0.119**
	RP	0.370	0.314	0.312	0.334	**0.382**	**0.408**	0.374	0.000	**0.000**	**0.000**	**0.000**	**0.000**	0.002	**0.527**	0.515	0.206	0.515	0.507	0.100	0.039	0.036
	RS	0.355	**0.359**	**0.327**	**0.392**	**0.382**	0.392	**0.395**	0.000	**0.000**	**0.000**	**0.000**	**0.000**	**0.003**	0.511	0.505	0.205	0.524	0.493	0.092	0.049	0.051
	Trunc	0.326	0.286	0.266	0.365	0.315	0.329	0.367	0.000	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.511	0.530	0.207	**0.582**	0.505	0.110	0.066	0.052
	AE	0.181	0.331	0.240	0.310	0.319	0.214	0.297	0.000	**0.000**	**0.000**	**0.000**	**0.000**	0.001	0.514	0.523	0.201	0.526	0.509	0.093	0.103	0.101
	GeoPres	**0.445**	0.270	0.263	0.227	0.351	0.344	0.318	**0.001**	**0.000**	**0.000**	**0.000**	**0.000**	0.002	0.507	0.568	0.209	0.528	**0.527**	**0.134**	**0.140**	0.114

Table 9: Raw task scores for `sentence-transformers/all-mpnet-base-v2`.