Why Mean Pooling Works: Quantifying Second-Order Collapse in Text Embeddings

Tomomasa Hara[1], Hiroto Kurita[1], Masaaki Imaizumi[2,3,4], Kentaro Inui[5,1,4], Sho Yokoi[6,1,4]
[1]Tohoku University, [2]The University of Tokyo, [3]Kyoto University, [4]RIKEN, [5]MBZUAI, [6]NINJAL
{hara.tomomasa.s8, hiroto.kurita.q4}@dc.tohoku.ac.jp
imaizumi@g.ecc.u-tokyo.ac.jp kentaro.inui@mbzuai.ac.ae yokoi@ninjal.ac.jp

Abstract

For constructing text embeddings, mean pooling, which averages token embeddings, is the standard approach. This paper examines whether mean pooling actually works well in real text encoders. First, we note that mean pooling can collapse information beyond the first-order statistics of the token embeddings, such as second-order statistics that capture their spatial structure, potentially mapping distinct token embedding distributions to similar text embeddings. Motivated by this concern, we propose a simple metric to quantify such a collapse induced by mean pooling. Then, using this metric, we empirically measure how often this collapse arises in actual models and texts, and find that mean pooling works well in modern text encoders. In particular, this collapse is less likely to arise in contrastive fine-tuned text encoders than in their pretrained backbone models. We also find that the robustness of these text encoders to collapse stems from the concentration of token embeddings within each text. In addition, we find that robustness to this collapse, as quantified by our proposed metric, correlates with downstream task performance. Overall, our findings help explain why modern text encoders remain effective despite relying on seemingly coarse mean pooling.

1 Introduction

Text embeddings, which represent sentences, paragraphs, and documents as single vectors, are used across a wide range of NLP tasks (Enevoldsen et al., 2025), including information retrieval (Thakur et al., 2021) and automatic evaluation (Rei et al., 2020). Text embeddings have also become essential for modern retrieval-augmented generation (Lewis et al., 2020) applications. Despite their widespread use, understanding text embeddings remains an important open challenge, as prior work has tackled this problem from the perspectives of geometry (Xiao et al., 2023) and dimensionality (Takeshita et al., 2025). This paper focuses on the aggregation method that constructs embeddings from token-level representations.

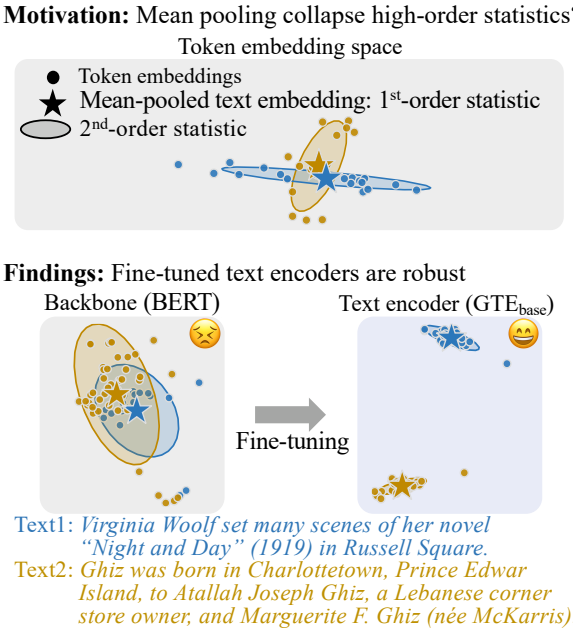


Figure 1: Overview of this work. **Top:** Mean pooling can map distinct token embedding distributions to similar text embeddings. This is because mean pooling summarizes distributions using only their first-order statistics, collapsing higher-order statistics. **Bottom:** We empirically find that modern fine-tuned text encoders are robust to such a collapse (§ 5). Each panel visualizes token and text embeddings for two texts, output by BERT (Devlin et al., 2019) and GTE$_{base}$ (Li et al., 2023) via PCA projection without centering. This example was discovered using the metric in § 4.

As such, the most standard aggregation method is *mean pooling*, which averages token embeddings. This simple aggregation has offered empirical advantages across embedding methods, from classical static token embeddings (Wieting et al., 2016; Shen et al., 2018) to modern contextualized embeddings from Transformer (Vaswani et al., 2017) encoders (Reimers and Gurevych, 2019). Indeed, state-of-the-art text encoders use mean pool-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 47180–47201

ing (Lee et al., 2025; Nussbaum et al., 2025).

While mean pooling is standard, does it work well in actual models? Figure 1 (top) illustrates a potential collapse mode: even when two texts yield distinct token embeddings in the embedding space ($[\bullet, \cdots, \bullet] \neq [\bullet, \cdots, \bullet]$), mean pooled text embeddings may become indistinguishable ($\bigstar \approx \bigstar$). This is because the mean-pooled text embedding corresponds only to the first-order statistic (mean) of the token embedding distribution. Thus, mean pooling does not capture the spatial structure of token embeddings, as reflected in second- and higher-order statistics, such as the covariance matrix. Turning to text embeddings via mean pooling, several prior works have proposed alternative approaches to mean-pooled text embeddings that represent a text as a list of token embeddings prior to mean pooling, such as optimal transport between token embedding lists (Kusner et al., 2015; Zhao et al., 2019; Yokoi et al., 2020; Lee et al., 2022) (§ 2.2). Nevertheless, mean-pooled text embeddings remain dominant in practice, as they are both computationally efficient and empirically effective.

This potential collapse raises the question: do actual models suffer from it, or do they avoid it? To address this question, we first propose a framework to evaluate the robustness of a text encoder to such a collapse. Specifically, we introduce the degree of **S**econd-**O**rder **C**ollapse by **M**ean pooling, a metric that quantifies the severity of the second-order statistics collapse by mean pooling (§ 4).

Next, using this proposed metric, we evaluate how well mean pooling works across actual models and texts. Our results empirically show that mean pooling in modern text encoders works well. In particular, we make the following findings: (i) We find that contrastive fine-tuned Transformer text encoders are less prone to collapse than their pretrained backbone models, suggesting that these fine-tuned encoders are robust to collapse (§ 5, Figure 1 bottom). (ii) We further find that this robustness stems from the concentration of token embeddings within each text, which arises through the Transformer layer (§ 6). (iii) We also observe that the proposed metric correlates with downstream task performance, suggesting that robustness to the collapse is one factor in the success of modern text encoders on downstream tasks (§ 7).

Overall, by rethinking mean pooling, which may appear to be a coarse aggregation method, this paper offers a new perspective on the effectiveness of modern text encoders.

2 Related Work

This paper focuses on the roughness of mean pooling in modern text embeddings, namely, its loss of higher-order statistics of token embeddings. Text encoders with mean pooling have become a standard paradigm (§ 2.1). Meanwhile, as an alternative paradigm, methods that preserve higher-order statistics without pooling have also been proposed (§ 2.2); yet modern text encoders achieve empirical performance comparable to these methods. We offer a new perspective on why modern text encoders work well by rethinking mean pooling.

2.1 Text Embeddings via Mean Pooling

In recent NLP, mean pooling over token embeddings from contrastive fine-tuned Transformer encoders has become the dominant paradigm for text embeddings (Reimers and Gurevych, 2019; Gao et al., 2021; Wang et al., 2024; Li et al., 2023; Nussbaum et al., 2025; Lee et al., 2025). This dominance is driven by its compatibility with contrastive fine-tuning (Gao et al., 2021) and computational efficiency, including low memory use (T. and T., 2024) and compatibility with approximate nearest neighbor search (Douze et al., 2025).

Beyond these engineering motivations, this paper supports the effectiveness of mean pooling from the perspective that it incurs little information loss in contrastive fine-tuned text encoders.

2.2 Text Representations Preserving Higher-Order Statistics

As an alternative paradigm to mean pooling, methods that handle text representations while preserving higher-order statistics have also been proposed. One approach is to work directly with the list of token embeddings before pooling. Distances of such lists have been computed using optimal transport, which measures the minimum cost of transporting one list to another (Kusner et al., 2015; Zhao et al., 2019; Yokoi et al., 2020; Lee et al., 2022). Similarly, ColBERT (Khattab and Zaharia, 2020; Santhanam et al., 2022) measured the distance of token embedding lists by matching each query token to its most similar document token and aggregating the resulting score. BERTScore (Zhang* et al., 2020) also aggregated token-level similarities into an F-score-like metric. As another direction, Sen2Pro (Shen et al., 2023) represents each text as a probability distribution over text embeddings by sampling multiple embeddings via MC dropout and

data augmentation. GaussCSE (Yoda et al., 2024) directly predicts first- and second-order statistics to represent each text as a Gaussian distribution.

Meanwhile, modern text encoders with mean pooling are not only computationally lighter than these methods, but also achieve comparable empirical performance (Lee et al., 2022). This paper investigates why text encoders work well even when representing texts via simple mean pooling.

3 Preliminaries

In this paper, we quantify how mean pooling, by using only the first-order statistics of the original token embeddings, collapses its second-order statistics (§ 4). In this section, we formally introduce token embeddings and these statistics.

We consider constructing text embeddings for two texts t_1 and t_2 using a model $\boldsymbol{f}$. Given t_i, $\boldsymbol{f}$ outputs a token embedding list $\boldsymbol{X}_i$:

$$\boldsymbol{X}_i := [\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,n_i}] = \boldsymbol{f}(t_i) \in \mathbb{R}^{d \times n_i}. \quad (1)$$

Here, $\boldsymbol{x}_{i,j} \in \mathbb{R}^d$ is the j-th token embedding in t_i, d is the embedding dimension, and n_i is the number of tokens in t_i.

Mean pooling averages token embeddings to construct a text embedding $\boldsymbol{\mu}(\boldsymbol{X}_i) \in \mathbb{R}^d$:

$$\boldsymbol{\mu}(\boldsymbol{X}_i) := \frac{1}{n_i} \sum_{j=1}^{n_i} \boldsymbol{x}_{i,j}. \quad (2)$$

This text embedding $\boldsymbol{\mu}(\boldsymbol{X}_i)$ corresponds to the first-order statistic of $\boldsymbol{X}_i$ when viewed as an empirical distribution in the embedding space.

Meanwhile, the token embedding distribution also has a second-order statistic $\boldsymbol{\Sigma}(\boldsymbol{X}_i) \in \mathbb{R}^{d \times d}$:

$$\boldsymbol{\Sigma}(\boldsymbol{X}_i) \quad (3)$$
$$:= \frac{1}{n_i} \sum_{j=1}^{n_i} (\boldsymbol{x}_{i,j} - \boldsymbol{\mu}(\boldsymbol{X}_i))(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}(\boldsymbol{X}_i))^\top.$$

This second-order statistic captures the spatial structure of token embeddings. For simplicity, we focus on it as the lowest-order statistic not retained by mean pooling.

Based on these definitions, we quantify the collapse induced by mean pooling (§ 4) and examine how often it arises in practice (§ 5).

4 Second-Order Collapse by Mean Pooling

In this section, we provide both intuitive and formal characterizations of collapse by mean pooling, based on first- and second-order statistics. We first provide an intuitive understanding of when this collapse occurs by examining the similarity of first- and second-order statistics (§ 4.1). Based on this intuition, we then introduce a metric to quantify the severity of this collapse (§ 4.2).

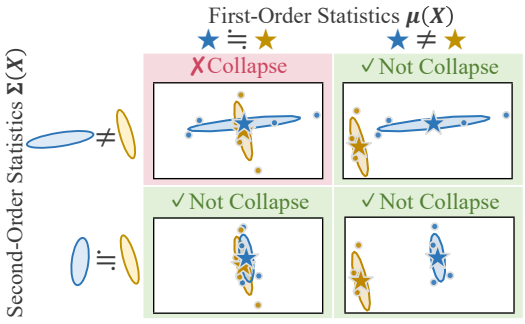


Figure 2: When mean pooling collapse arises (red) and does not (green). Mean pooling collapse occurs when the first-order statistics are similar while the second-order statistics differ. Each panel depicts a conceptual distribution in a two-dimensional embedding space, characterized by its first- and second-order statistics.

4.1 When Does Collapse Arise?

As discussed in § 1, mean pooling may collapse two distinct token embedding distributions into similar text embeddings. We characterize whether this collapse arises based on the similarity of their first-order statistics and that of their second-order statistics, as shown in Figure 2.

When Collapse Arises The collapse arises when first-order statistics are similar but second-order ones differ, i.e., $\boldsymbol{\mu}(\boldsymbol{X}_1) \approx \boldsymbol{\mu}(\boldsymbol{X}_2) \wedge \boldsymbol{\Sigma}(\boldsymbol{X}_1) \neq \boldsymbol{\Sigma}(\boldsymbol{X}_2)$. In this case, distributions that are distinct up to the second-order statistics may collapse into similar representations after mean pooling.

When Collapse Does Not Arise We also consider when the collapse does not arise. One case is when first-order statistics differ, i.e., $\boldsymbol{\mu}(\boldsymbol{X}_1) \neq \boldsymbol{\mu}(\boldsymbol{X}_2)$. Here, the distributions are separated by their means unless the variance is excessively large. Another case is when both first- and second-order statistics are similar, i.e., $\boldsymbol{\mu}(\boldsymbol{X}_1) \approx \boldsymbol{\mu}(\boldsymbol{X}_2)$ and $\boldsymbol{\Sigma}(\boldsymbol{X}_1) \approx \boldsymbol{\Sigma}(\boldsymbol{X}_2)$. Here, the distributions are similar, as reflected in their means.

4.2 Quantify the Severity of Collapse

Based on these intuitions, we introduce the degree of **S**econd-**O**rder **C**ollapse by **M**ean pooling (hereafter referred to as **SOCM**), a metric that quantifies the collapse of second-order statistics by mean

pooling. As discussed in § 4.1, whether the collapse by mean pooling arises depends on the similarity of first-order statistics and the similarity of second-order statistics. Thus, we define SOCM using the distance of the first-order statistics and the distance of the second-order statistics (§ 4.2.1). Specifically, we decompose the distance of token embedding distributions into first- and second-order components and construct SOCM from them (§ 4.2.2). We also design SOCM to satisfy desirable properties such as boundary conditions and monotonicity (§ 4.2.3).

4.2.1 Definition of SOCM

Given two token embedding lists $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, SOCM quantifies the severity of the collapse when applying mean pooling to them. Specifically, SOCM is defined as:

$$\mathrm{SOCM}(d_\mu, d_\Sigma) := (1 - d_\mu)d_\Sigma. \tag{4}$$

Here, d_μ is the distance of the first-order statistics $\boldsymbol{\mu}(\boldsymbol{X}_1)$ and $\boldsymbol{\mu}(\boldsymbol{X}_2)$, and d_Σ is the distance of the second-order statistics $\boldsymbol{\Sigma}(\boldsymbol{X}_1)$ and $\boldsymbol{\Sigma}(\boldsymbol{X}_2)$.

d_μ is the scaled squared Euclidean distance:

$$d_\mu := \|\boldsymbol{\mu}(\boldsymbol{X}_1) - \boldsymbol{\mu}(\boldsymbol{X}_2)\|_2^2/4. \tag{5}$$

We assume unit-norm means, $\|\boldsymbol{\mu}(\boldsymbol{X}_i)\| = 1$, as is common in the use cases of text embeddings (Enevoldsen et al., 2025). Under this normalization, $d_\mu \in [0, 1]$[1] and corresponds to a common distance for text embeddings (Enevoldsen et al., 2025).

d_Σ is defined as the scaled Bures Wasserstein distance (Dowson and Landau, 1982):

$$d_\Sigma := \mathrm{tr}\big(\boldsymbol{\Sigma}(\boldsymbol{X}_1) + \boldsymbol{\Sigma}(\boldsymbol{X}_2) - 2\big(\boldsymbol{\Sigma}(\boldsymbol{X}_1)^{1/2}\boldsymbol{\Sigma}(\boldsymbol{X}_2)\boldsymbol{\Sigma}(\boldsymbol{X}_1)^{1/2}\big)^{1/2}\big)/4. \tag{6}$$

We assume $\mathrm{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)) \leq 2$ to put d_Σ in the same range as d_μ. This assumption corresponds to the scenario described in § 4.1, where the variance of token embeddings does not become excessively large. Under this assumption, $d_\Sigma \in [0, 1]$[2], matching the range of d_μ.

Since $d_\mu, d_\Sigma \in [0, 1]$ as above, Eq. (4) implies $\mathrm{SOCM} \in [0, 1]$. This SOCM value indicates a more severe collapse for larger values. As discussed in § 4.1, the collapse arises when first-order statistics are similar (d_μ is small), but second-order statistics differ (d_Σ is large). SOCM reflects this intuition via $(1 - d_\mu)d_\Sigma$.

In the following subsections, we discuss the validity of this SOCM design.

4.2.2 Decomposing Distributional Distance

We adopt d_μ and d_Σ as the first- and second-order distances, respectively. This choice follows from decomposing the distance of token embedding distributions into first- and second-order components.

Decomposition d_μ and d_Σ correspond to the first- and second-order components of the L_2-Wasserstein distance W_2^2 of token embedding distributions (Dowson and Landau, 1982):

$$W_2^2\big(\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}_1), \boldsymbol{\Sigma}(\boldsymbol{X}_1)), \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}_2), \boldsymbol{\Sigma}(\boldsymbol{X}_2))\big)/4 = d_\mu + d_\Sigma. \tag{7}$$

For simplicity, we characterize each token embedding list $\boldsymbol{X}_i$ as a Gaussian $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}_i), \boldsymbol{\Sigma}(\boldsymbol{X}_i))$. This means that higher-order moments are not explicitly distinguished, though the Gaussian characterization is motivated by computational tractability and stability in Wasserstein-based evaluation (see Appendix A for details).

Motivation for W_2^2 This decomposition of the Wasserstein distance is a natural choice for quantifying the severity of the collapse by mean pooling. The Wasserstein distance is a widely used metric for measuring the distance between distributions, considering not only first- but also higher-order statistics (Villani, 2009). Indeed, optimal transport, which measures semantic similarity over token embedding lists (discussed in § 2.2), is based on this Wasserstein distance (Peyré and Cuturi, 2020). Therefore, d_μ and d_Σ, which decompose the Wasserstein distance, are natural components for constructing SOCM.

4.2.3 Validity of SOCM Form

We adopt Eq. (4) as SOCM. To verify its validity, we enumerate desirable properties for quantifying the severity of the collapse and show that this form satisfies all of them.

Desirable Properties To quantify the collapse severity, we define five desirable properties:

(a) When Collapse Arises:
$d_\mu = 0 \land d_\Sigma = 1 \Leftrightarrow \mathrm{SOCM} = 1$.

(b) When Collapse Does Not Arise:
$d_\mu = 1 \lor d_\Sigma = 0 \Leftrightarrow \mathrm{SOCM} = 0$.

[1] A factor $1/4$ rescales d_μ to $[0, 1]$.

[2] As with d_μ, the factor $1/4$ rescales d_Σ to $[0, 1]$ under the trace bound.

(c) Monotonicity in d_μ: $\dfrac{\partial\,\mathrm{SOCM}}{\partial d_\mu} \leq 0.$

(d) Monotonicity in d_Σ: $\dfrac{\partial\,\mathrm{SOCM}}{\partial d_\Sigma} \geq 0.$

(e) Interaction of d_μ and d_Σ: $\dfrac{\partial^2\,\mathrm{SOCM}}{\partial d_\mu\,\partial d_\Sigma} \leq 0.$

(a) When Collapse Arises Property (a) requires that SOCM $= 1$ if and only if the collapse arises in the extreme case: $d_\mu = 0 \wedge d_\Sigma = 1$. As discussed in § 4.1, collapse arises when first-order statistics are similar but second-order statistics differ. This extreme case corresponds to identical first-order statistics ($d_\mu = 0$) and maximally different second-order statistics ($d_\Sigma = 1$).

(b) When Collapse Does Not Arise Property (b) requires that SOCM $= 0$ if and only if the collapse does not arise: $d_\mu = 1 \vee d_\Sigma = 0$. As discussed in § 4.1, collapse does not arise when first-order statistics differ or when both first- and second-order statistics are similar. $d_\mu = 1 \vee d_\Sigma = 0$ indicates these cases. Specifically, $d_\mu = 1$ indicates maximally different first-order statistics. Additionally, $d_\Sigma = 0$ indicates identical second-order statistics, so similarity is determined by d_μ.

(c) Monotonicity in d_μ Property (c) requires that SOCM is monotonically non-increasing in d_μ. This reflects that the collapse is less severe as the first-order statistics become more distant.

(d) Monotonicity in d_Σ Property (d) requires that SOCM is monotonically non-decreasing in d_Σ. This reflects that the collapse becomes more severe as the second-order statistics become more distant.

(e) Interaction between d_μ and d_Σ Property (e) requires that the impact of d_Σ on SOCM decreases as d_μ increases. While property (d) requires SOCM to increase with d_Σ, this increase should be smaller for large d_μ. When first-order statistics are already distant (large d_μ), they capture distributional differences, so second-order differences matter less. The mixed derivative enforces this: as d_μ increases, SOCM becomes less sensitive to d_Σ.

Form of SOCM Satisfying the Properties The adopted form in Eq. (4) satisfies all desirable properties (a)–(e) (see Appendix B for the proof). Figure 3 plots values of SOCM over d_μ and d_Σ, confirming properties (a)–(e). The four corners of the plot correspond to the scenarios in Figure 2. SOCM is high when collapse arises and low otherwise.

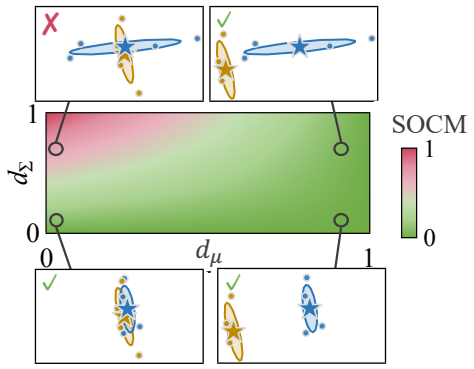


Figure 3: SOCM values for each combination of (d_μ, d_Σ). The figure confirms that SOCM satisfies all properties (a)–(e). The four corners correspond to the scenarios in Figure 2. SOCM is higher when mean pooling collapse occurs (top-left) and lower when it does not (the other three cases).

5 Experiment

Using SOCM, we empirically measured how often the collapse induced by mean pooling arose across actual models and texts. We found that contrastive fine-tuned Transformer text encoders tend to be less prone to the collapse than their pretrained backbones.

5.1 Experimental Procedure and Setting

Overview To examine how often the collapse arises, we prepared a text pair dataset $D = \{(t_1, t_2)\}$ and a model $\boldsymbol{f}$. Given a text t_i, we obtain a token embedding list $\boldsymbol{X}_i$ from the model $\boldsymbol{f}$. We computed SOCM for each token embedding list pair $(\boldsymbol{X}_1, \boldsymbol{X}_2)$.

Normalization of Token Embedding Lists We normalized each token embedding list. The definition of SOCM in § 4.2.1 assumes this normalization, $\|\boldsymbol{\mu}(\boldsymbol{X})\| = 1$. This assumption aligns with standard practice in text embedding applications, where a unit-normalized embedding $\boldsymbol{\mu}(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|$ is used (Enevoldsen et al., 2025). To satisfy this assumption, given a token embedding list $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_i}] \in \mathbb{R}^{d \times n_i}$ from model $\boldsymbol{f}$, we use the normalized version $\boldsymbol{X}_i^{\mathrm{norm}}$ defined as:

$$\boldsymbol{X}_i^{\mathrm{norm}} = \left[\frac{\boldsymbol{x}_{i,1}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2}, \ldots, \frac{\boldsymbol{x}_{i,n_i}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2}\right] \in \mathbb{R}^{d \times n_i}. \tag{8}$$

$\boldsymbol{\mu}(\boldsymbol{X}_i^{\mathrm{norm}}) = \boldsymbol{\mu}(\boldsymbol{X}_i)/\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|$ holds under this normalization (see Appendix C). We then computed SOCM for each pair $(\boldsymbol{X}_1^{\mathrm{norm}}, \boldsymbol{X}_2^{\mathrm{norm}})$.

Model	Avg. SOCM ↓
BERT	0.396
→ Unsup-SimCSE-mean	**0.193** (−0.203)
→ $E5_{base}$	**0.029** (−0.367)
→ GTE_{base}	**0.018** (−0.378)
MiniLM	0.242
→ all-MiniLM-L12-v2	0.313 (+0.071)
→ $E5_{small}$	**0.099** (−0.143)
→ GTE_{small}	**0.055** (−0.187)
MPNet	0.117
→ all-mpnet-base-v2	**0.100** (−0.017)
nomic-bert-2048	0.139
→ nomic-embed-text-v1.5	**0.122** (−0.017)

Table 1: Average SOCM values for each model on text pairs from Wikipedia. For text encoders derived from backbone models, values in parentheses show the change in SOCM. Bold values indicate a reduction.

Dataset We constructed a dataset of text pairs from Wikipedia (Gao et al., 2021), which contains 1 million texts. Specifically, we randomly sampled 1,000 texts and generated 499,500 text pairs by comparing them pairwise. Experiments on another dataset, MS MARCO (Bajaj et al., 2018), led to similar conclusions (Appendix E).

Model We examined popular Transformer-based text encoders that use mean pooling. Specifically, we used the following models, each obtained by contrastive fine-tuning the corresponding backbone: mean pooling-based Unsupervised SimCSE (Gao et al., 2021), $E5_{base}$ (Wang et al., 2024), and GTE_{base} (Li et al., 2023) (BERT (Devlin et al., 2019)); all-MiniLM-L12-v2 (Reimers and Gurevych, 2019), $E5_{small}$ (Wang et al., 2024), and GTE_{small} (Li et al., 2023) (MiniLM (Wang et al., 2020)); all-mpnet-base-v2 (Reimers and Gurevych, 2019) (MPNet (Song et al., 2020)); and nomic-embed-text-v1.5 (Nussbaum et al., 2025) (nomic-bert-2048 (Nussbaum et al., 2025)). We also applied SOCM to the corresponding backbones.

5.2 Results

Quantitative Analysis Table 1 shows the average SOCM for each model on Wikipedia. Overall, contrastive fine-tuned text encoders tended to have lower SOCM than their backbone models. This indicates that contrastive fine-tuned text encoders are less prone to collapse by mean pooling in practice.

Qualitative Analysis We further validated our findings by visualizing embedding spaces. The bottom of Figure 1 shows 2D PCA projections without centering for two semantically unrelated texts: "*Virginia Woolf set many scenes of her novel "Night and Day" (1919) in Russell Square*" and "*Ghiz was born in Charlottetown, Prince Edward Island, to Atallah Joseph Ghiz, a Lebanese corner store owner, and Marguerite F. Ghiz (née McKarris).*". For this example, BERT exhibits high SOCM $= 0.618$ while GTE_{base} shows low SOCM $= 0.024$.[3] For BERT, the token embedding distributions differed in spread but had similar means. In contrast, for GTE_{base}, the means separated the two token embedding distributions. These visualizations are consistent with the quantitative analyses.

In summary, these results suggest that contrastive fine-tuned text encoders are more robust to the collapse by mean pooling. This suggests that mean pooling, despite appearing coarse, works well in contrastive fine-tuned text encoders.

6 How Do Fine-Tuned Text Encoders Avoid Collapse?

In this section, we investigate how fine-tuned text encoders avoid collapse induced by mean pooling. To explain this behavior, we focus on the concentration of token embeddings within each text, as illustrated for GTE_{base} in Figure 1 (bottom). Specifically, we theoretically show that token embeddings concentrate under certain conditions in a simplified Transformer formulation, and that such concentration reduces SOCM. We then empirically examine whether fine-tuned text encoders behave consistently with this theoretical account.

6.1 Theoretical Study

In the following, we formally show that token embedding concentration within a text, induced by internal components such as the attention matrix and the residual connection, leads to lower SOCM.

To analyze how Transformer-layer components contribute to token embedding concentration, we introduce a simplified formulation based on a single-head self-attention block. This formulation abstracts the layer into three components: self-attention with projection, residual connection, and a per-token transformation.

Assumption 1 (Setting). *Let* $\boldsymbol{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n] \in \mathbb{R}^{d \times n}$ *denote the input token embeddings. The self-*

[3] OurTool values are computed in the original 768-dimensional space before dimensionality reduction for visualization.

attention and projection components produce the attention output $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{W}^o \boldsymbol{W}^v \boldsymbol{H} \boldsymbol{A}^\top \in \mathbb{R}^{d \times n}, \tag{9}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the attention weight matrix after softmax, and $\boldsymbol{W}^v \in \mathbb{R}^{d_v \times d}$ and $\boldsymbol{W}^o \in \mathbb{R}^{d \times d_v}$ are the value and output projection matrices, respectively.[4] *The residual connection adds $\boldsymbol{Z}$ to the input $\boldsymbol{H}$, yielding the post-residual embeddings $\boldsymbol{Y}$:*

$$\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{H} \in \mathbb{R}^{d \times n}. \tag{10}$$

Finally, a per-token transformation g, which abstracts post-attention operations such as LayerNorm and FFN, is applied to each $\boldsymbol{y}_i$ to produce the final token embeddings $\boldsymbol{X}$:

$$\boldsymbol{x}_i = g(\boldsymbol{y}_i), \quad \boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}. \tag{11}$$

We assume $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{\eta}, c\boldsymbol{I}_d) \quad (\boldsymbol{\eta} \neq \boldsymbol{0})$ and treat $\boldsymbol{A}$ as fixed for analytical tractability. We also define the spread of any matrix $\boldsymbol{M} = [\boldsymbol{m}_1, \cdots, \boldsymbol{m}_n]$ to quantify the degree of token embedding concentration:

$$S(\boldsymbol{M}) \coloneqq \frac{1}{n} \sum_{j=1}^{n} \|\boldsymbol{m}_j - \mu(\boldsymbol{M})\|_2^2. \tag{12}$$

Based on this formulation, token embedding concentration can arise under the following three conditions: (i) the attention matrix and projection matrices reduce the spread of the attention-branch output $\boldsymbol{Z}$, (ii) the residual connection does not destroy the concentration introduced before it, and (iii) the per-token transformation g does not excessively disperse the residual output. We formalize these conditions as follows.

Definition 1 (Ratio of spread contraction through $\boldsymbol{A}$, $\boldsymbol{W}^v$, and $\boldsymbol{W}^o$)**.** *Let $\boldsymbol{P} = \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top \in \mathbb{R}^{n \times n}$, and define*

$$\lambda := \|\boldsymbol{W}^o \boldsymbol{W}^v\|_{\text{op}}^2 \frac{\|\boldsymbol{P}\boldsymbol{A}\|_F^2}{n-1}. \tag{13}$$

Intuition. This λ relates the spread of the attention-branch output $\boldsymbol{Z}$ to that of the input $\boldsymbol{H}$; a smaller λ indicates that $\boldsymbol{Z}$ is more concentrated across tokens.

Definition 2 (Ratio of token spread in input $\boldsymbol{H}$ to residual output scale $\|\mu(\boldsymbol{Y})\|$)**.** *Define*

$$r := \frac{\mathbb{E}_{\boldsymbol{H}}\left[S(\boldsymbol{H})\right]}{\mathbb{E}_{\boldsymbol{H}}\left[\|\mu(\boldsymbol{Y})\|_2^2\right]}. \tag{14}$$

[4]For simplicity, we omit bias terms from $\boldsymbol{W}^v$ and $\boldsymbol{W}^o$.

Intuition. This quantity compares the spread of the input token $\boldsymbol{H}$ with the scale of the residual output $\boldsymbol{Y}$.

Definition 3 (Relative spread ratio of $\boldsymbol{X}$ to $\boldsymbol{Y}$ through transformation g)**.** *Let $C > 0$ denote the smallest constant such that*

$$\frac{\mathbb{E}_{\boldsymbol{H}}\left[S(\boldsymbol{X})\right]}{\mathbb{E}_{\boldsymbol{H}}\left[\|\mu(\boldsymbol{X})\|_2^2\right]} \leq C \frac{\mathbb{E}_{\boldsymbol{H}}\left[S(\boldsymbol{Y})\right]}{\mathbb{E}_{\boldsymbol{H}}\left[\|\mu(\boldsymbol{Y})\|_2^2\right]}. \tag{15}$$

Intuition. This quantity measures how much the per-token transformation g amplifies the relative spread of token embeddings.

Theorem 1 (Token embeddings become concentrated within each text)**.** *Using Definitions 1–3, if $\lambda < 1$, then the normalized spread of the final token embeddings satisfies*

$$\frac{\mathbb{E}_{\boldsymbol{H}}\left[S(\boldsymbol{X})\right]}{\mathbb{E}_{\boldsymbol{H}}\left[\|\mu(\boldsymbol{X})\|_2^2\right]} = O(rC) \quad (r, C \to 0). \tag{16}$$

Intuition. When $\lambda < 1$, if r and C are also small, $\mathbb{E}_{\boldsymbol{H}}[S(\boldsymbol{X})]/\mathbb{E}_{\boldsymbol{H}}[\|\mu(\boldsymbol{X})\|_2^2]$ becomes small, meaning that token embeddings concentrate.

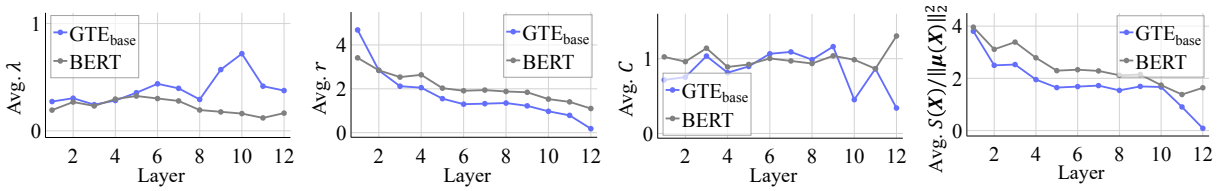
Sketch of proof. $\boldsymbol{Z}$ is concentrated after passing through the attention and projection (captured by λ); the influence of the spread of $\boldsymbol{H}$ on the residual output $\boldsymbol{Y}$ becomes relatively small (captured by r); and g does not disperse $\boldsymbol{Y}$ (captured by C), so $\boldsymbol{X}$ remains concentrated. A detailed proof is provided in Appendix F. □

Having established conditions for token embedding concentration, we now connect this concentration result to SOCM.

Theorem 2 (Token embedding concentration leads to low SOCM)**.** *Let $\boldsymbol{f}$ be a model that outputs token embeddings $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,n_i}] = \boldsymbol{f}(t_i)$ for input texts t_1 and t_2. Suppose that, for $i = 1, 2$, $\frac{S(\boldsymbol{X}_i)}{\|\mu(\boldsymbol{X}_i)\|_2^2} < \varepsilon$. Then, the SOCM value for $(\boldsymbol{X}_1^{\text{norm}}, \boldsymbol{X}_2^{\text{norm}})$ (Eq. (8)) satisfies* $\text{SOCM} = O(\varepsilon) \quad (\varepsilon \to 0)$.

Intuition. That is, when $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ is small, meaning that token embeddings concentrate within each text, SOCM also becomes small.

Sketch of proof. When $\boldsymbol{X}$ is concentrated, the token embeddings are nearly identical within each text, so d_Σ is small, and hence SOCM is small. A detailed proof is provided in Appendix F. □



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 4: Layer-wise analysis of token embedding concentration for BERT and GTE$_{\text{base}}$ on the Wikipedia dataset. (a) Avg. λ: a smaller value indicates that the attention-branch output $\boldsymbol{Z}$ is more concentrated across tokens relative to the input $\boldsymbol{H}$. (b) Avg. r: the spread of the input $\boldsymbol{H}$ relative to the scale of the residual output $\boldsymbol{Y}$. (c) Avg. C: how much the per-token transformation g amplifies the relative spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: a smaller value indicates that token embeddings concentrate within each text.

6.2 Empirical Study

In the theoretical study, we showed that token embedding concentration leads to lower SOCM, where concentration is characterized by λ for self-attention and projection, r for the residual connection, and C for the per-token transformation. In the following, we empirically examine each of these quantities in fine-tuned text encoders.

Setting We use the Wikipedia dataset from § 5. In the following, we compare BERT and GTE$_{\text{base}}$ (see Appendix H for results on other models).

Self-Attention and Projection (λ) We verify whether $\lambda < 1$ holds in practice. Figure 4a shows the layer-wise average of λ for BERT and GTE$_{\text{base}}$. In both models, $\lambda < 1$ held across all layers, and the values were broadly comparable between the two models.

Residual Connection (r) We examine the empirical behavior of r, which compares the spread of the input token embeddings with the scale of the residual output. Figure 4b shows the layer-wise average of $r = \frac{S(\boldsymbol{H})}{\|\mu(\boldsymbol{Y})\|_2^2}$ for BERT and GTE$_{\text{base}}$. In GTE$_{\text{base}}$, r was lower than in BERT across layers and decreased toward zero in the final layer.

Per-Token Transformation (C) We examine the empirical behavior of C, which measures the degree to which the per-token transformation g disperses token embeddings. Figure 4c shows the layer-wise average of $C := \frac{S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2}{S(\boldsymbol{Y})/\|\mu(\boldsymbol{Y})\|_2^2}$ for BERT and GTE$_{\text{base}}$. C was broadly similar in scale across both models, though GTE$_{\text{base}}$ showed smaller values in some later layers.

Token Embedding Concentration We examine the concentration measure $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ appearing in Theorem 1. Figure 4d shows the layer-wise average of $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ for BERT and GTE$_{\text{base}}$. In GTE$_{\text{base}}$, this value was lower than in BERT, particularly in the later layers. This is consistent with Theorem 1, given the observed values of λ, r, and C. This concentration pattern is also consistent with the lower SOCM observed for GTE$_{\text{base}}$, as implied by Theorem 2. Looking more closely at the components, r decreased toward zero in the final layers of GTE$_{\text{base}}$, where $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ also decreased toward zero. By contrast, λ was broadly comparable between GTE$_{\text{base}}$ and BERT, suggesting that the degree to which the attention and projection matrices concentrate token embeddings is similar across the two models. These observations suggest that the lower SOCM in fine-tuned text encoders is related less to a stronger concentrating effect in the attention and projection branch itself, and more to the reduced relative influence of input spread in the residual output, as captured by r.

Why Concentration Occurs in Fine-Tuned Encoders One possible explanation for why concentration arises in fine-tuned encoders is that contrastive fine-tuning applies supervision at the level of the mean-pooled text embedding. In this setting, contrastive learning does not directly supervise how individual token embeddings should be arranged, but instead optimizes them through their contribution to the resulting text embedding. Because contrastive learning encourages this embedding to remain discriminative (Wang and Isola, 2020), representations are more useful when discriminative properties are reflected stably in the average. One way to achieve this is for token embeddings within a text to become more concentrated around their mean, so that such properties are more directly captured by the mean-pooled embedding. From this perspective, the attention component may promote

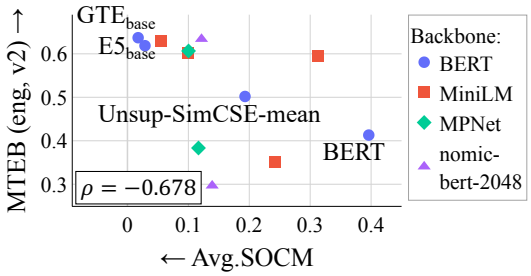


Figure 5: Scatter plot of average SOCM and MTEB (eng, v2) score. Each point represents a model. Marker types indicate the backbones. Models with a BERT backbone are annotated with their names.

concentration by mixing information across tokens, while the residual connection may help maintain this concentration through subsequent layers.

Connection to Prior Work These results are consistent with prior work on the geometry of token embeddings in fine-tuned text encoders (Xiao et al., 2023). Xiao et al. (2023) reported that contrastive fine-tuning leads to anisotropic token embeddings within each text, particularly in the later layers. Our findings connect this observation to robustness to collapse by mean pooling.

7 Correlation with Downstream Task Performance

In this section, we examine whether robustness to collapse by mean pooling, as quantified by SOCM, correlates with downstream task performance.

Setting As a downstream task, we used MTEB (eng, v2) (Enevoldsen et al., 2025), a standard text embedding evaluation benchmark consisting of 41 tasks. For each model used in § 5, we compared its MTEB score with the SOCM in Table 1.

Result Figure 5 shows a scatter plot of the average SOCM against the average MTEB score for each model. We observed a negative correlation (Spearman's $\rho = -0.678$, $p = 0.015$), indicating that models with lower SOCM tended to achieve higher downstream task performance. This result suggests that robustness to collapse by mean pooling may be one contributing factor to the success of modern text encoders on downstream tasks.

Discussion We further compare SOCM with $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$, the token concentration measure introduced in § 6, by examining their correlations with downstream task performance. Table 2

	ρ
SOCM	$\mathbf{-0.678}$
$S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$	-0.622

Table 2: Spearman's ρ between MTEB (eng, v2) scores and each of SOCM and $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$.

shows that SOCM is more strongly negatively correlated with downstream task performance than $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$. One possible explanation is that $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ can be small even when semantically dissimilar texts yield similar mean-pooled embeddings, as it does not capture inter-text separation. SOCM, by contrast, also decreases when d_μ increases, i.e., when the mean vectors of two token embedding distributions are far apart, thereby capturing this separation. Since contrastive fine-tuning pushes negative pairs apart (Wang et al., 2020), it may increase d_μ and thus lower SOCM. Further investigation into how contrastive fine-tuning improves downstream task performance remains an important direction for future work.

8 Conclusion

This paper offered a new perspective on why modern text encoders remain effective despite adopting seemingly coarse mean pooling. First, we argued that mean pooling can discard second-order statistics, causing distinct token embedding distributions to collapse into similar text embeddings. We then proposed a metric to quantify this collapse (§ 4). Using this metric, we empirically found that contrastive fine-tuned text encoders are less prone to collapse than their pretrained backbones (§ 5) in practice. We also showed that token embedding concentration within each text underlies the robustness of fine-tuned text encoders against this collapse (§ 6). We further observed that this metric correlates with downstream task performance (§ 7).

Future Work A fuller mathematical account of why contrastive fine-tuning reduces SOCM and improves downstream performance remains an important direction. SOCM may also be useful for encoder development; one concrete direction is to use it as a regularization term during training. Another direction is to extend this analysis to LLM-based generation and reasoning, for example, by examining context compression by mean pooling (Feldman and Artzi, 2025).

Limitations

This work has several limitations.

Restriction to Second-order Statistics This paper approximates the information in token embedding lists using only first- and second-order moments. Specifically, we approximate token embedding lists with Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}_i), \boldsymbol{\Sigma}(\boldsymbol{X}_i))$, which are fully characterized by their first- and second-order moments. However, the actual distribution of token embedding lists may also contain information from third-order and higher moments. Analyzing third-order and higher moments in token embedding lists remains an interesting direction for future work.

Assumptions in the Proposed Metric In defining the proposed metric SOCM, we impose the assumptions $\|\boldsymbol{\mu}(\boldsymbol{X}_i)\| = 1$ and $\mathrm{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)) \leq 2$. As shown in Appendix J, these assumptions hold under the normalization procedure in § 5 and certain conditions on model architecture (LayerNorm with shared parameters across dimensions and sufficient similarity of token embeddings within texts). However, the proposed metric may not work effectively in cases where these assumptions are violated, for instance, when the second-order moment $\mathrm{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i))$ becomes extremely large. Extending the generality of the metric to broader settings remains future work.

Toward Theoretical Understanding In § 6, we partially addressed the theoretical understanding of the observed robustness. Specifically, we showed theoretically that token embedding concentration leads to lower SOCM, and empirically verified that fine-tuned text encoders satisfy the conditions under which such concentration occurs. However, why contrastive fine-tuning induces token embedding concentration in the first place remains an open question. A deeper theoretical investigation of this mechanism is an interesting direction for future work.

Practical Utility of SOCM In this paper, we use SOCM as an analysis tool for characterizing robustness to collapse by mean pooling across models and texts. An interesting direction for future work is to explore practical uses of SOCM beyond analysis, such as incorporating it into training objectives as a regularization term.

Scope Limited to Text Embeddings This paper focuses on text embeddings and does not address LLM-based applications, such as generation or reasoning. Broadening the scope to these LLM-based applications, for example, by investigating how context compression by mean pooling influences generation or reasoning, remains an interesting direction for future work.

Focus on Mean Pooling This paper focuses on mean pooling as the target aggregation method. Mean pooling remains the dominant pooling strategy in modern text encoders. However, richer pooling methods, such as SIF weighting (Arora et al., 2017), have also been proposed as alternatives to simple averaging. Investigating such methods is an interesting direction for future work.

Ethical Consideration

This study analyzes embeddings using models released under the MIT License (MiniLM, E5_{small}, E5_{base}, $\text{GTE}_{\text{small}}$, GTE_{base}, Unsupervised SimCSE, MPNet) and Apache License 2.0 (all-MiniLM-L12-v2, all-mpnet-base-v2, nomic-bert-2048, nomic-embed-text-v1.5). Our analysis, which examines the embedding representations output by these models for given input texts, falls within their intended use cases. We use datasets (Wikipedia, MS MARCO, MTEB (eng, v2)) released under Apache License 2.0. These datasets are used as provided without additional preprocessing to remove social biases, personal information, or offensive content, and thus may reflect various biases present in the original data.

Acknowledgments

This work was supported by AMED Grant Number JP26wm0625405, JSPS KAKENHI Grant Number JP22H05106, the JST FOREST Program Grant Number JPMJFR2331, and JST BOOST Grant Number JPMJBS2421. We would like to thank the members of the Tohoku NLP Group, as well as the organizers and participants of IBIS 2025 and NLP 2026, for their insightful and encouraging feedback.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations (ICLR)*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, An-

drew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *Preprint*, arXiv:1611.09268.

Juan Antonio Cuesta-Albertos, C Matrán-Bea, and A Tuero-Dı´az. 1996. On lower bounds for the L2-Wasserstein metric in a Hilbert space. *Journal of Theoretical Probability*, 9:263–283.

Marco Cuturi. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems 26 (NIPS)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss library. *Preprint*, arXiv:2401.08281.

D.C Dowson and B.V Landau. 1982. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Veysel Çağatan, and 63 others. 2025. MMTEB: Massive Multilingual Text Embedding Benchmark. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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 55–65.

Yair Feldman and Yoav Artzi. 2025. Simple Context Compression: Mean-Pooling and Multi-Ratio Training. *Preprint*, arXiv:2510.20797.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems 30 (NIPS)*, volume 30. Curran Associates, Inc.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 957–966.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, and 28 others. 2025. Gemini Embedding: Generalizable Embeddings from Gemini. *Preprint*, arXiv:2503.07891.

Seonghyeon Lee, Dongha Lee, Seongbo Jang, and Hwanjo Yu. 2022. Toward Interpretable Semantic Textual Similarity via Optimal Transport-based Contrastive Sentence Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5969–5979.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33 (NIPS)*, pages 9459–9474.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *Preprint*, arXiv:2308.03281.

Zach Nussbaum, John Xavier Morris, Andriy Mulyar, and Brandon Duderstadt. 2025. Nomic Embed: Training a Reproducible Long Context Text Embedder. *Transactions on Machine Learning Research (TMLR)*.

Gabriel Peyré and Marco Cuturi. 2020. Computational Optimal Transport. *Preprint*, arXiv:1803.00567.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

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.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 3715–3734.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–450.

Lingfeng Shen, Haiyun Jiang, Lemao Liu, and Shuming Shi. 2023. Sen2Pro: A Probabilistic Perspective to Sentence Embedding from Pre-trained Language Model. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP)*, pages 315–333.

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 33 (NIPS)*, pages 16857–16867.

Hai Nguyen T. and Huong Le T. 2024. Enhancing ColBERT: A Method for Reducing Space Complexity and Accelerating Retrieval Speed. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation (PALIC)*, pages 820–829.

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 (EMNLP)*, pages 27705–27726.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems (NIPS) Datasets and Benchmarks Track (Round 2)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30 (NIPS)*.

Cedric Villani. 2009. *Optimal Transport: Old and New*, 1 edition, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *Preprint*, arXiv:2212.03533.

Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 9929–9939.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33 (NIPS)*, pages 5776–5788.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards Universal Paraphrastic Sentence Embeddings. In *International Conference on Learning Representations (ICLR)*.

Chenghao Xiao, Yang Long, and Noura Al Moubayed. 2023. On Isotropy, Contextualization and Learning Dynamics of Contrastive-based Sentence Representation Learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12266–12283.

Shohei Yoda, Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2024. Sentence Representations via Gaussian Embedding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 418–425.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word Rotator's Distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations (ICLR)*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. 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 563–578.

A Gaussian Characterization

As noted in the § 4, our method evaluates the target distributions by approximating them with Gaussian distributions. This is an intentional design choice motivated by computational tractability and stability in Wasserstein-based evaluation.

Advantages of the Gaussian Characterization First, when using the Wasserstein distance, the Gaussian characterization offers substantial advantages despite ignoring higher-order moments. For general non-Gaussian distributions, computing the Wasserstein distance typically requires costly linear programming-based algorithms whose complexity scales poorly with dimensionality, making them impractical in high dimensions (Cuturi, 2013). From this perspective, the Gaussian characterization is essential for efficient computation in high-dimensional spaces, and its computational and analytical benefits outweigh the limitation of not explicitly modeling higher-order moments.

Information Captured by First- and Second-order Statistics Furthermore, focusing on first- and second-order statistics lets us efficiently capture much of the information in the data distribution. In particular, it is known that the L_2-Wasserstein distance computed using only up to second-order statistics provides a universal lower bound on the true L_2-Wasserstein distance between arbitrary distributions (Cuesta-Albertos et al., 1996). In this sense, introducing a Gaussian characterization is not merely a heuristic simplification; rather, it recovers information that is theoretically guaranteed in terms of the Wasserstein distance up to second-order statistics. Moreover, such characterizations are widely used in practice. For example, the Fréchet Inception Distance (FID), a standard metric for evaluating generative models, also relies on the L_2-Wasserstein distance between Gaussian approximations characterized by first- and second-order statistics, and has been successfully used across a broad range of distributions (Heusel et al., 2017).

Taken together, these theoretical guarantees and empirical precedents support the use of second-order statistics-based Wasserstein characterization as a principled and practically effective approach.

B Proof that SOCM Satisfies Desirable Properties

In this section, we prove that SOCM as defined in Eq. (4) satisfies the desirable properties (a)–(e) introduced in § 4.2.3. Specifically, we show that

$$\mathrm{SOCM}(d_\mu, d_\Sigma) = (1 - d_\mu)d_\Sigma \tag{17}$$

satisfies all properties (a)–(e):

(a) $d_\mu = 0 \wedge d_\Sigma = 1 \Leftrightarrow \mathrm{SOCM} = 1$

(b) $d_\mu = 1 \vee d_\Sigma = 0 \Leftrightarrow \mathrm{SOCM} = 0$

(c) $\dfrac{\partial\,\mathrm{SOCM}}{\partial d_\mu} \leq 0$

(d) $\dfrac{\partial\,\mathrm{SOCM}}{\partial d_\Sigma} \geq 0$

(e) $\dfrac{\partial^2\,\mathrm{SOCM}}{\partial d_\mu\,\partial d_\Sigma} \leq 0$

B.1 Proof of Property (a)

We prove property (a): $d_\mu = 0 \wedge d_\Sigma = 1 \Leftrightarrow \mathrm{SOCM} = 1$. We first show $(d_\mu = 0 \wedge d_\Sigma = 1 \Rightarrow \mathrm{SOCM} = 1)$. When $d_\mu = 0$ and $d_\Sigma = 1$,

$$\mathrm{SOCM}(0, 1) = (1 - 0) \cdot 1 \tag{18}$$
$$= 1. \tag{19}$$

Therefore, $d_\mu = 0 \wedge d_\Sigma = 1 \Rightarrow \mathrm{SOCM} = 1$ holds. Next, we show $(\mathrm{SOCM} = 1 \Rightarrow d_\mu = 0 \wedge d_\Sigma = 1)$. Assume $\mathrm{SOCM}(d_\mu, d_\Sigma) = 1$, i.e.,

$$(1 - d_\mu)d_\Sigma = 1. \tag{20}$$

Since $d_\mu, d_\Sigma \in [0, 1]$, we have $(1 - d_\mu) \in [0, 1]$ and $d_\Sigma \in [0, 1]$. Therefore, the above equation holds only when $1 - d_\mu = 1$ and $d_\Sigma = 1$, which implies $d_\mu = 0$ and $d_\Sigma = 1$. Therefore, $\mathrm{SOCM} = 1 \Rightarrow d_\mu = 0 \wedge d_\Sigma = 1$ holds.

B.2 Proof of Property (b)

We prove property (b): $d_\mu = 1 \vee d_\Sigma = 0 \Leftrightarrow \mathrm{SOCM} = 0$. We first show $(d_\mu = 1 \vee d_\Sigma = 0 \Rightarrow \mathrm{SOCM} = 0)$. When $d_\mu = 1$,

$$\mathrm{SOCM}(1, d_\Sigma) = (1 - 1) \cdot d_\Sigma \tag{21}$$
$$= 0. \tag{22}$$

Also, when $d_\Sigma = 0$,

$$\mathrm{SOCM}(d_\mu, 0) = (1 - d_\mu) \cdot 0 \tag{23}$$
$$= 0. \tag{24}$$

Therefore, $d_\mu = 1 \vee d_\Sigma = 0 \Rightarrow \text{SOCM} = 0$ holds. Next, we show $(\text{SOCM} = 0 \Rightarrow d_\mu = 1 \vee d_\Sigma = 0)$. Assume $\text{SOCM}(d_\mu, d_\Sigma) = 0$, i.e.,

$$(1 - d_\mu)d_\Sigma = 0. \tag{25}$$

This holds when $1 - d_\mu = 0$ or $d_\Sigma = 0$, i.e., when $d_\mu = 1$ or $d_\Sigma = 0$. Therefore, $\text{SOCM} = 0 \Rightarrow d_\mu = 1 \vee d_\Sigma = 0$ holds.

B.3 Proof of Property (c)

We prove property (c): $\frac{\partial\,\text{SOCM}}{\partial d_\mu} \leq 0$. Taking the partial derivative of $\text{SOCM}(d_\mu, d_\Sigma) = (1 - d_\mu)d_\Sigma$ with respect to d_μ,

$$\frac{\partial\,\text{SOCM}}{\partial d_\mu} = -d_\Sigma. \tag{26}$$

Since $d_\Sigma \in [0, 1]$, we have $-d_\Sigma \leq 0$. Therefore, $\frac{\partial\,\text{SOCM}}{\partial d_\mu} \leq 0$ holds.

B.4 Proof of Property (d)

We prove property (d): $\frac{\partial\,\text{SOCM}}{\partial d_\Sigma} \geq 0$. Taking the partial derivative of $\text{SOCM}(d_\mu, d_\Sigma) = (1 - d_\mu)d_\Sigma$ with respect to d_Σ,

$$\frac{\partial\,\text{SOCM}}{\partial d_\Sigma} = 1 - d_\mu. \tag{27}$$

Since $d_\mu \in [0, 1]$, we have $1 - d_\mu \geq 0$. Therefore, $\frac{\partial\,\text{SOCM}}{\partial d_\Sigma} \geq 0$ holds.

B.5 Proof of Property (e)

We prove property (e): $\frac{\partial^2\,\text{SOCM}}{\partial d_\mu\,\partial d_\Sigma} \leq 0$. From the proof of property (c),

$$\frac{\partial\,\text{SOCM}}{\partial d_\mu} = -d_\Sigma. \tag{28}$$

Taking the partial derivative with respect to d_Σ,

$$\frac{\partial^2\,\text{SOCM}}{\partial d_\mu\,\partial d_\Sigma} = -1 \leq 0. \tag{29}$$

Therefore, property (e) holds.

B.6 Generalization

More generally, for any $\alpha > 0$ and $\beta > 0$, the following form also satisfies all properties (a)–(e):

$$\text{SOCM}_{\alpha,\beta}(d_\mu, d_\Sigma) = (1 - d_\mu)^\alpha d_\Sigma^\beta. \tag{30}$$

The proof follows analogously, replacing $1 - d_\mu$ and d_Σ with their respective powers. Our adopted form in Eq. (4) corresponds to the simple case $\alpha = \beta = 1$.

C Proof of Normalization Property

We prove that under the normalization defined in § 5, the relationship $\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \boldsymbol{\mu}(\boldsymbol{X}_i)/\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|$ holds.

Statement Given a token embedding list $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_i}] \in \mathbb{R}^{d \times n_i}$ and its normalized version

$$\boldsymbol{X}_i^{\text{norm}} = \left[\frac{\boldsymbol{x}_{i,1}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}, \ldots, \frac{\boldsymbol{x}_{i,n_i}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}\right] \in \mathbb{R}^{d \times n_i}, \tag{31}$$

we show that $\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \boldsymbol{\mu}(\boldsymbol{X}_i)/\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|$.

Proof By the definition of mean pooling, we have

$$\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \frac{1}{n_i}\sum_{j=1}^{n_i} \frac{\boldsymbol{x}_{i,j}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}. \tag{32}$$

Since $\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|$ is a scalar independent of the summation index j, we can factor it out:

$$\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \frac{1}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|} \cdot \frac{1}{n_i}\sum_{j=1}^{n_i} \boldsymbol{x}_{i,j}. \tag{33}$$

The remaining summation is precisely the definition of $\boldsymbol{\mu}(\boldsymbol{X}_i)$:

$$\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \frac{1}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|} \cdot \boldsymbol{\mu}(\boldsymbol{X}_i) \tag{34}$$

$$= \frac{\boldsymbol{\mu}(\boldsymbol{X}_i)}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}. \tag{35}$$

Furthermore, this implies that $\|\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}})\| = 1$, which is the desired property for computing SOCM as defined in § 4.2.1.

D Implementation Details

This section provides detailed information about the implementation of our experiments described in § 5, § 6, and § 7.

D.1 Dataset Details

Preprocessing For comparison across different models and datasets, we did not use any task-specific prefixes (e.g., `query:`, `passage:`) when encoding texts. Note that some text embedders are designed to utilize such prefixes to distinguish between different text types (Wang et al., 2024; Li et al., 2023).

Dataset URLs Table 3 shows the URLs for the datasets used in our experiments.

Dataset	URL
Wikipedia	https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/resolve/main/wiki1m_for_simcse.txt
MSMARCO	https://huggingface.co/datasets/mteb/msmarco
MSMARCO Hard Negatives	https://huggingface.co/datasets/sentence-transformers/msmarco-co-condenser-margin-mse-sym-mnrl-mean-v1
MTEB (eng, v2)	https://github.com/embeddings-benchmark/mteb

Table 3: URLs for datasets used in our experiments.

Language In our experiments, we used English-language datasets.

D.2 Model Details

Table 4 lists the specific model identifiers from Hugging Face for each model used in our experiments. For most models, we used publicly available pre-trained checkpoints. For Unsupervised SimCSE (Gao et al., 2021), we trained the model ourselves using the original codebase, as the unsupervised checkpoint was not publicly available at the time of our experiments. We followed the default training configuration provided in the official implementation, using BERT-base-uncased as the backbone model and training on English Wikipedia with the default hyperparameters specified in the original paper.

E Additional Dataset Results for § 5

To validate the generalizability of our findings, we conducted additional experiments on the MS-MARCO.

Experimental Setting We followed the same experimental procedure as described in § 5. Specifically, we randomly sampled 1,000 texts from MS-MARCO passages and generated 499,500 text pairs by comparing these texts pairwise. We computed SOCM values for the same set of models used in § 5. In addition, we computed SOCM for 50,000 query–negative passage pairs from MS MARCO hard negatives.

Results Table 5 shows the average SOCM values for each model on MSMARCO, and Table 6 shows those on the MS MARCO hard negatives. In both cases, we observed similar trends to the Wikipedia results (§ 5).

F Proof of Theorem 1

Recall that for any matrix $\boldsymbol{M} \in \mathbb{R}^{d\times n}$,

$$S(\boldsymbol{M}) = \frac{1}{n}\sum_{j=1}^{n} \|\boldsymbol{m}_j - \mu(\boldsymbol{M})\|_2^2 = \frac{1}{n}\|\boldsymbol{M}\boldsymbol{P}\|_F^2, \tag{36}$$

where $\boldsymbol{P} = \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top$ is the centering matrix.

We first control the spread of $\boldsymbol{Z} = \boldsymbol{W}^o\boldsymbol{W}^v\boldsymbol{H}\boldsymbol{A}^\top$. Since $\boldsymbol{P}$ is symmetric,

$$\boldsymbol{Z}\boldsymbol{P} = \boldsymbol{W}^o\boldsymbol{W}^v\boldsymbol{H}\boldsymbol{A}^\top\boldsymbol{P} = \boldsymbol{W}^o\boldsymbol{W}^v\boldsymbol{H}(\boldsymbol{P}\boldsymbol{A})^\top. \tag{37}$$

Hence, by the operator norm inequality $\|\boldsymbol{A}\boldsymbol{B}\|_F \le \|\boldsymbol{A}\|_{\mathrm{op}}\|\boldsymbol{B}\|_F$,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Z})] &= \frac{1}{n}\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{Z}\boldsymbol{P}\|_F^2\right] \\ &= \frac{1}{n}\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{W}^o\boldsymbol{W}^v\boldsymbol{H}(\boldsymbol{P}\boldsymbol{A})^\top\|_F^2\right] \\ &\le \frac{\|\boldsymbol{W}^o\boldsymbol{W}^v\|_{\mathrm{op}}^2}{n}\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{H}(\boldsymbol{P}\boldsymbol{A})^\top\|_F^2\right].\end{aligned} \tag{38}$$

Write $\boldsymbol{H} = \boldsymbol{\eta}\boldsymbol{1}^\top + \boldsymbol{E}$, where the columns of $\boldsymbol{E}$ are i.i.d. $\mathcal{N}(\boldsymbol{0}, c\boldsymbol{I}_d)$. Since each row of $\boldsymbol{A}$ sums to one (i.e., $\boldsymbol{A}\boldsymbol{1} = \boldsymbol{1}$), we have $(\boldsymbol{P}\boldsymbol{A})\boldsymbol{1} = \boldsymbol{P}(\boldsymbol{A}\boldsymbol{1}) = \boldsymbol{P}\boldsymbol{1} = \boldsymbol{0}$, and thus $\boldsymbol{1}^\top(\boldsymbol{P}\boldsymbol{A})^\top = \boldsymbol{0}^\top$. Therefore, the mean part vanishes:

$$\boldsymbol{H}(\boldsymbol{P}\boldsymbol{A})^\top = \boldsymbol{E}(\boldsymbol{P}\boldsymbol{A})^\top. \tag{39}$$

By the isotropy of the Gaussian noise, $\mathbb{E}_{\boldsymbol{h}}\left[\|\boldsymbol{E}\boldsymbol{B}\|_F^2\right] = cd\,\|\boldsymbol{B}\|_F^2$ for any $\boldsymbol{B} \in \mathbb{R}^{n\times n}$. Taking $\boldsymbol{B} = (\boldsymbol{P}\boldsymbol{A})^\top$,

$$\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Z})] \le \frac{\|\boldsymbol{W}^o\boldsymbol{W}^v\|_{\mathrm{op}}^2}{n} \cdot cd\,\|\boldsymbol{P}\boldsymbol{A}\|_F^2. \tag{40}$$

On the other hand, since $\|\boldsymbol{P}\|_F^2 = \mathrm{tr}(\boldsymbol{P}) = n-1$,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})] &= \frac{1}{n}\mathbb{E}_{\boldsymbol{h}}[\|\boldsymbol{E}\boldsymbol{P}\|_F^2] \\ &= \frac{cd}{n}\|\boldsymbol{P}\|_F^2 \\ &= \frac{cd(n-1)}{n}.\end{aligned} \tag{41}$$

Model	model_name	Params (M)
MiniLM	microsoft/MiniLM-L12-H384-uncased	30
all-MiniLM-L12-v2	sentence-transformers/all-MiniLM-L12-v2	30
$E5_{small}$	intfloat/e5-small-v2	30
GTE_{small}	thenlper/gte-small	30
BERT	bert-base-uncased	110
Unsup-SimCSE-mean	h-tomo/unsup-simcse-bert-base-uncased-mean	110
$E5_{base}$	intfloat/e5-base-v2	110
GTE_{base}	thenlper/gte-base	110
MPNet	microsoft/mpnet-base	109
all-mpnet-base-v2	sentence-transformers/all-mpnet-base-v2	109
nomic-bert-2048	nomic-ai/nomic-bert-2048	137
nomic-embed-text-v1.5	nomic-ai/nomic-embed-text-v1.5	137

Table 4: Details of the Hugging Face models used in our experiments, with parameter counts (in millions) taken from the corresponding backbone model sizes.

Model	Avg. SOCM ↓
BERT	0.491
→ Unsup-SimCSE-mean	**0.269** (−0.222)
→ $E5_{base}$	**0.043** (−0.448)
→ GTE_{base}	**0.025** (−0.466)
MiniLM	0.289
→ all-MiniLM-L12-v2	**0.348** (+0.059)
→ $E5_{small}$	**0.085** (−0.204)
→ GTE_{small}	**0.055** (−0.234)
MPNet	0.106
→ all-mpnet-base-v2	**0.094** (−0.012)
nomic-bert-2048	0.110
→ nomic-embed-text-v1.5	**0.133** (+0.023)

Table 5: Average SOCM values for each model on text pairs from MS MARCO passages. For text encoders derived from backbone models, values in parentheses show the change in SOCM. Bold values indicate a reduction.

Model	Avg. SOCM ↓
BERT	0.480
→ Unsup-SimCSE-mean	**0.257** (−0.224)
→ $E5_{base}$	**0.036** (−0.445)
→ GTE_{base}	**0.017** (−0.464)
MiniLM	0.340
→ all-MiniLM-L12-v2	**0.330** (−0.010)
→ $E5_{small}$	**0.087** (−0.253)
→ GTE_{small}	**0.048** (−0.292)
MPNet	0.129
→ all-mpnet-base-v2	**0.093** (−0.036)
nomic-bert-2048	0.131
→ nomic-embed-text-v1.5	**0.104** (−0.027)

Table 6: Average SOCM values for each model on query–negative passage pairs from MS MARCO hard negatives. For text encoders derived from backbone models, values in parentheses show the change in SOCM. Bold values indicate a reduction.

Combining the above two inequalities,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Z})] &\le \|\boldsymbol{W}^o\boldsymbol{W}^v\|_{\mathrm{op}}^2 \frac{\|\boldsymbol{P}\boldsymbol{A}\|_F^2}{n-1}\,\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})] \\ &= \lambda\,\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})]. \end{aligned} \tag{42}$$

By Definition 1 and the hypothesis $\lambda < 1$,

Next, we control the spread of $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{H}$. Using $\boldsymbol{Y}\boldsymbol{P} = \boldsymbol{Z}\boldsymbol{P} + \boldsymbol{H}\boldsymbol{P}$ and the Minkowski inequality,

$$\begin{aligned}\sqrt{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Y})]} &= \frac{1}{\sqrt{n}}\Big(\mathbb{E}_{\boldsymbol{h}}\big[\|\boldsymbol{Y}\boldsymbol{P}\|_F^2\big]\Big)^{1/2} \\ &\le \frac{1}{\sqrt{n}}\Big(\mathbb{E}_{\boldsymbol{h}}\big[\|\boldsymbol{Z}\boldsymbol{P}\|_F^2\big]\Big)^{1/2} \\ &\quad + \frac{1}{\sqrt{n}}\Big(\mathbb{E}_{\boldsymbol{h}}\big[\|\boldsymbol{H}\boldsymbol{P}\|_F^2\big]\Big)^{1/2} \\ &= \sqrt{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Z})]} + \sqrt{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})]} \\ &\le (1+\sqrt{\lambda})\sqrt{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})]}. \end{aligned} \tag{43}$$

Squaring both sides yields

$$\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Y})] \le (1+\sqrt{\lambda})^2\,\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})]. \tag{44}$$

Therefore,

$$\frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Y})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{Y})\|_2^2]} \le (1+\sqrt{\lambda})^2 \frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{H})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{Z}+\boldsymbol{H})\|_2^2]}. \tag{45}$$

By Definition 2, the right-hand side equals r, hence

$$\frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Y})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{Y})\|_2^2]} \leq (1+\sqrt{\lambda})^2 r = O(r), \quad (46)$$

where the hidden constant depends only on λ.

Finally, Definition 3 gives

$$\begin{aligned}\frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{X})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{X})\|_2^2]} &\leq C\,\frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{Y})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{Y})\|_2^2]} \\ &\leq C(1+\sqrt{\lambda})^2 r. \quad (47)\end{aligned}$$

We therefore conclude that

$$\frac{\mathbb{E}_{\boldsymbol{h}}[S(\boldsymbol{X})]}{\mathbb{E}_{\boldsymbol{h}}[\|\mu(\boldsymbol{X})\|_2^2]} = O(rC) \quad (r, C \to 0). \quad (48)$$

This completes the proof.

G Proof of Theorem 2

Let t_1, t_2 be arbitrary texts, and let $\boldsymbol{X}_i = \boldsymbol{f}(t_i)$ for $i = 1, 2$. By definition of $\boldsymbol{\mu}(\cdot)$,

$$\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}}) = \frac{\boldsymbol{\mu}(\boldsymbol{X}_i)}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2}, \quad (49)$$

hence $\|\boldsymbol{\mu}(\boldsymbol{X}_i^{\text{norm}})\|_2 = 1$. Since normalization scales each token embedding by the scalar $1/\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2$, the covariance matrix scales quadratically:

$$\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}}) = \frac{1}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2^2}\boldsymbol{\Sigma}(\boldsymbol{X}_i). \quad (50)$$

Using $S(\boldsymbol{X}) = \operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}))$, the assumption gives

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) = \frac{S(\boldsymbol{X}_i)}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|_2^2} < \varepsilon \qquad (i = 1, 2). \quad (51)$$

Since $(1 - d_\mu) \leq 1$, we have

$$SOCM(\boldsymbol{X}_1^{\text{norm}}, \boldsymbol{X}_2^{\text{norm}}) \leq d_\Sigma. \quad (52)$$

By definition of d_Σ,

$$\begin{aligned}d_\Sigma = \frac{1}{4}\operatorname{tr}\Big(&\boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}}) + \boldsymbol{\Sigma}(\boldsymbol{X}_2^{\text{norm}}) \\ &- 2\Big(\boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}})^{1/2}\boldsymbol{\Sigma}(\boldsymbol{X}_2^{\text{norm}}) \\ &\qquad \boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}})^{1/2}\Big)^{1/2}\Big). \quad (53)\end{aligned}$$

Since $\left(\boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}})^{1/2}\boldsymbol{\Sigma}(\boldsymbol{X}_2^{\text{norm}})\boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}})^{1/2}\right)^{1/2}$ is positive semidefinite, its trace is nonnegative, and therefore

$$d_\Sigma \leq \frac{1}{4}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_1^{\text{norm}})) + \frac{1}{4}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_2^{\text{norm}})) < \frac{\varepsilon}{2}. \quad (54)$$

Hence,

$$SOCM(\boldsymbol{X}_1^{\text{norm}}, \boldsymbol{X}_2^{\text{norm}}) \leq d_\Sigma < \frac{\varepsilon}{2}, \quad (55)$$

which gives $SOCM(\boldsymbol{X}_1^{\text{norm}}, \boldsymbol{X}_2^{\text{norm}}) = O(\varepsilon)$ as $\varepsilon \to 0$. This completes the proof.

H Additional Results for § 6

H.1 Results for Additional Models

We provide additional results for the layer-wise analysis of token embedding concentration described in § 6. In addition to the results for BERT and GTE_{base} reported in § 6, Figures 6–10 show the layer-wise averages of λ, r, C, and $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ for the remaining model pairs: E5_{base} and BERT (Figure 6), Unsupervised SimCSE and BERT (Figure 7), $\text{GTE}_{\text{small}}$ and MiniLM (Figure 8), E5_{small} and MiniLM (Figure 9), and all-MiniLM-L12-v2 and MiniLM (Figure 10). These results tend to show similar trends to those observed for BERT and GTE_{base} in § 6.

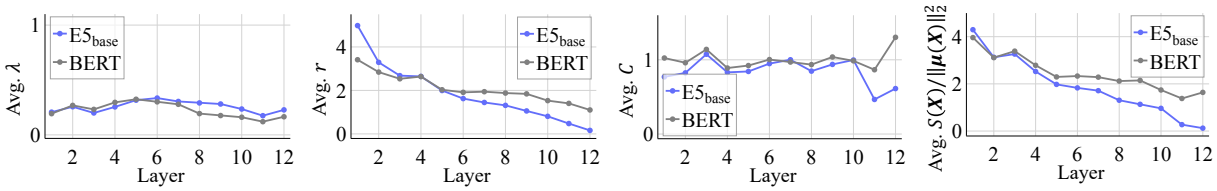
H.2 Within-Text Token Embedding Concentration

In § 6, we connected our findings to the anisotropy of token embeddings within each text reported by Xiao et al. (2023). As anisotropy is commonly measured using cosine similarity (Xiao et al., 2023), we quantify within-text token embedding concentration via the layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j, \boldsymbol{x}_k)$, where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ denotes the token embeddings of a text, and each average is taken over all texts in the dataset.[5] A higher value indicates greater concentration, consistent with higher within-text anisotropy as described by Xiao et al. (2023). Figures 11–16 show these results for all model pairs. Fine-tuned text encoders tend to show higher within-text average cosine similarity than their backbone counterparts, particularly in the later layers. This trend is consistent with the lower SOCM and smaller $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$ observed in fine-tuned text encoders (§ 6), supporting the connection between within-text token embedding concentration and robustness against collapse by mean pooling.

I Analysis of d_μ and d_Σ

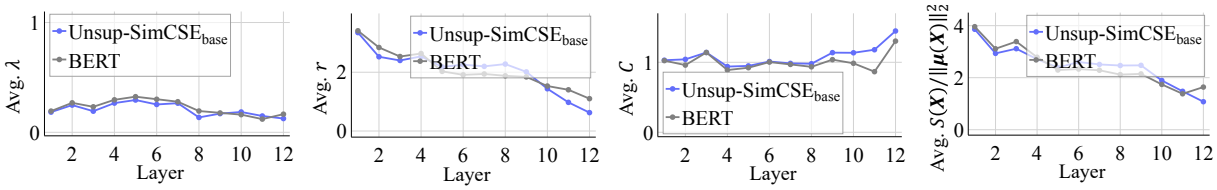
We examine the components d_μ and d_Σ that constitute SOCM to understand the variation in SOCM across model pairs observed in § 5.

[5] Note that this quantity is not identical to the anisotropy measure used by Xiao et al. (2023).



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 6: Layer-wise analysis of token embedding concentration for BERT and E5_{base} on the Wikipedia dataset. (a) Avg. λ: ratio of spread contraction through $\boldsymbol{A}$, $\boldsymbol{W}^v$, and $\boldsymbol{W}^o$. (b) Avg. r: ratio of token spread in input $\boldsymbol{H}$ to residual output scale $\|\mu(\boldsymbol{Y})\|$. (c) Avg. C: how much the per-token transformation g amplifies the normalized spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: the normalized spread of token embeddings within each text.



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 7: Layer-wise analysis of token embedding concentration for BERT and Unsupervised SimCSE on the Wikipedia dataset. (a) Avg. λ: a smaller value indicates that the attention-branch output $\boldsymbol{Z}$ is more concentrated across tokens relative to the input $\boldsymbol{H}$. (b) Avg. r: the spread of the input $\boldsymbol{H}$ relative to the scale of the residual output $\boldsymbol{Y}$. (c) Avg. C: how much the per-token transformation g amplifies the relative spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: a smaller value indicates that token embeddings concentrate within each text.

Observations Figure 17 shows scatter plots of d_μ and d_Σ for all examined models on the Wikipedia dataset. Across all model pairs, fine-tuned text encoders tend to exhibit small d_Σ values.

Connection to Token Embedding Concentration This tendency is consistent with the results of § 6, which showed that fine-tuned text encoders tend to concentrate token embeddings around their within-text mean. When token embeddings concentrate, their within-text covariance $\boldsymbol{\Sigma}(\boldsymbol{X})$ becomes small. This directly suppresses d_Σ: since d_Σ is the scaled Bures-Wasserstein distance between $\boldsymbol{\Sigma}(\boldsymbol{X}_1)$ and $\boldsymbol{\Sigma}(\boldsymbol{X}_2)$, it satisfies $d_\Sigma \leq \big(\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_1)) + \text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_2))\big)/4$. Therefore, when both $\boldsymbol{\Sigma}(\boldsymbol{X}_1)$ and $\boldsymbol{\Sigma}(\boldsymbol{X}_2)$ are small, d_Σ is bounded to be small as well, regardless of how different the two covariance matrices are from each other.

J Trace Bound under Normalization

In this section, we prove that the assumption $\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)) \leq 2$ in § 4.2.1 holds under the normalization procedure described in § 5 and certain assumptions about the model architecture.

J.1 Setup and Notation

We consider token embeddings output from a LayerNorm layer (Devlin et al., 2019). Let $\boldsymbol{y}_{i,j} \in \mathbb{R}^d$ denote the hidden state before LayerNorm for the j-th token in text i. The k-th dimension of the token embedding after LayerNorm is given by:

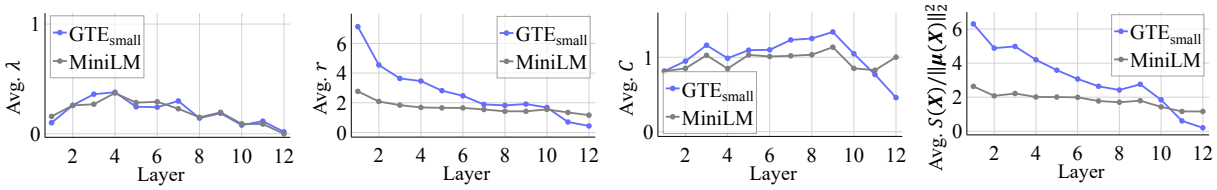
$$x_{i,j,k} = \gamma_k \cdot \frac{y_{i,j,k} - m_{i,j}}{s_{i,j}} + \beta_k, \tag{56}$$

where $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters for dimension k, and $m_{i,j}$ and $s_{i,j}$ are the mean and standard deviation computed across dimensions:

$$m_{i,j} = \frac{1}{d}\sum_{k=1}^{d} y_{i,j,k}, \tag{57}$$

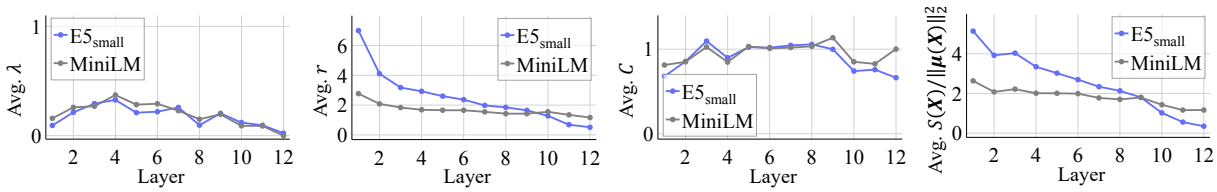
$$s_{i,j} = \sqrt{\frac{1}{d}\sum_{k=1}^{d}(y_{i,j,k} - m_{i,j})^2}. \tag{58}$$

Let $\boldsymbol{x}_{i,j} = [x_{i,j,1}, \ldots, x_{i,j,d}]^\top \in \mathbb{R}^d$ denote the token embedding after LayerNorm. The token embedding list before normalization is $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_i}]$ with $\boldsymbol{\mu}(\boldsymbol{X}_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} \boldsymbol{x}_{i,j}$. After normalization by the mean norm (as described in § 5), we obtain the normalized token embedding



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 8: Layer-wise analysis of token embedding concentration for MiniLM and GTE$_{\text{small}}$ on the Wikipedia dataset. (a) Avg. λ: a smaller value indicates that the attention-branch output $\boldsymbol{Z}$ is more concentrated across tokens relative to the input $\boldsymbol{H}$. (b) Avg. r: the spread of the input $\boldsymbol{H}$ relative to the scale of the residual output $\boldsymbol{Y}$. (c) Avg. C: how much the per-token transformation g amplifies the relative spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: a smaller value indicates that token embeddings concentrate within each text.



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 9: Layer-wise analysis of token embedding concentration for MiniLM and E5$_{\text{small}}$ on the Wikipedia dataset. (a) Avg. λ: ratio of spread contraction through $\boldsymbol{A}$, $\boldsymbol{W}^v$, and $\boldsymbol{W}^o$. (b) Avg. r: ratio of token spread in input $\boldsymbol{H}$ to residual output scale $\|\mu(\boldsymbol{Y})\|$. (c) Avg. C: how much the per-token transformation g amplifies the normalized spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: the normalized spread of token embeddings within each text.

list $\boldsymbol{X}_i^{\text{norm}} = [\boldsymbol{x}_{i,1}^{\text{norm}}, \ldots, \boldsymbol{x}_{i,n_i}^{\text{norm}}]$, where:

$$\boldsymbol{x}_{i,j}^{\text{norm}} = \frac{\boldsymbol{x}_{i,j}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}. \tag{59}$$

J.2 Assumptions

We make the following assumptions:

Assumption 1 LayerNorm parameters are shared across all dimensions, i.e., $\gamma_1 = \cdots = \gamma_d = \gamma$ and $\beta_1 = \cdots = \beta_d = \beta$. This assumption reflects the initialization of LayerNorm parameters in typical Transformer models (Devlin et al., 2019), where $\gamma_1 = \cdots = \gamma_d = 1$ and $\beta_1 = \cdots = \beta_d = 0$.

Assumption 2 Token embeddings within the same text exhibit sufficient similarity. Specifically, the expected cosine similarity between token embeddings within the same text satisfies $\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,k})] \geq 1/3$. This assumption is justified by empirical observations that contextualized embeddings within the same text are anisotropic (Ethayarajh, 2019).

J.3 Proof

Under Assumptions 1 and 2, we show that $\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) \leq 2$ holds for the normalized token embedding list $\boldsymbol{X}_i^{\text{norm}}$.

Relating normalized and unnormalized covariance matrices We first establish that:

$$\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) = \frac{1}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2}\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)), \tag{60}$$

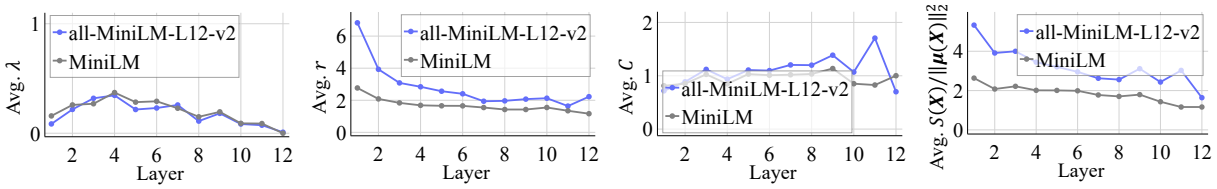
where $\boldsymbol{\Sigma}(\boldsymbol{X}_i) = \frac{1}{n_i}\sum_{j=1}^{n_i}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}(\boldsymbol{X}_i))(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}(\boldsymbol{X}_i))^\top$. This follows from the definition $\boldsymbol{x}_{i,j}^{\text{norm}} = \frac{\boldsymbol{x}_{i,j}}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|}$ and the properties of the trace operator.

Computing the trace of unnormalized covariance Under Assumption 1, the norm of each token embedding after LayerNorm is constant:

$$\|\boldsymbol{x}_{i,j}\|^2 = \sum_{k=1}^{d} x_{i,j,k}^2 = d(\gamma^2 + \beta^2). \tag{61}$$

Using this fact and the relation $\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)) = \frac{1}{n_i}\sum_{j=1}^{n_i}\|\boldsymbol{x}_{i,j}\|^2 - \|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2$, we obtain:

$$\text{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i)) = d(\gamma^2 + \beta^2) - \|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2. \tag{62}$$



(a) Avg. λ. (b) Avg. r. (c) Avg. C. (d) Avg. $S(\boldsymbol{X})/\|\boldsymbol{\mu}(\boldsymbol{X})\|_2^2$.

Figure 10: Layer-wise analysis of token embedding concentration for MiniLM and all-MiniLM-L12-v2 on the Wikipedia dataset. (a) Avg. λ: ratio of spread contraction through $\boldsymbol{A}$, $\boldsymbol{W}^v$, and $\boldsymbol{W}^o$. (b) Avg. r: ratio of token spread in input $\boldsymbol{H}$ to residual output scale $\|\mu(\boldsymbol{Y})\|$. (c) Avg. C: how much the per-token transformation g amplifies the normalized spread of token embeddings. (d) Avg. $S(\boldsymbol{X})/\|\mu(\boldsymbol{X})\|_2^2$: the normalized spread of token embeddings within each text.

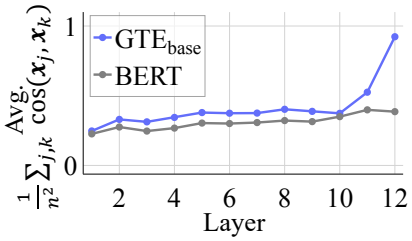


Figure 11: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j,\boldsymbol{x}_k)$ for BERT and GTE$_{\text{base}}$ on the Wikipedia dataset.

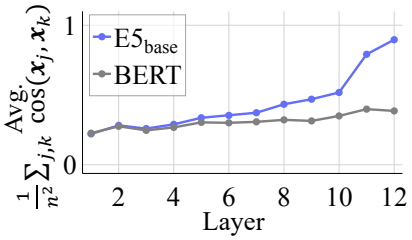


Figure 12: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j,\boldsymbol{x}_k)$ for BERT and E5$_{\text{base}}$ on the Wikipedia dataset.

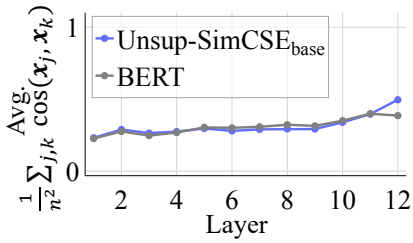


Figure 13: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j,\boldsymbol{x}_k)$ for BERT and Unsupervised SimCSE on the Wikipedia dataset.

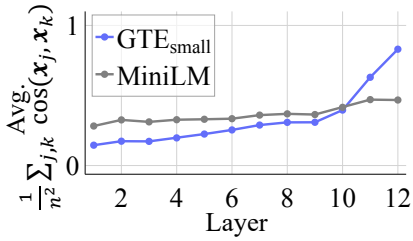


Figure 14: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j,\boldsymbol{x}_k)$ for MiniLM and GTE$_{\text{small}}$ on the Wikipedia dataset.

Computing the squared norm of the mean The squared norm of the mean can be expressed as:

$$\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2 = \frac{1}{n_i^2}\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\boldsymbol{x}_{i,j}^\top\boldsymbol{x}_{i,k} \tag{63}$$

$$= \frac{1}{n_i^2}\left(\sum_{j=1}^{n_i}\|\boldsymbol{x}_{i,j}\|^2 + 2\sum_{j<k}\boldsymbol{x}_{i,j}^\top\boldsymbol{x}_{i,k}\right) \tag{64}$$

$$= \frac{1}{n_i^2}\Big(n_i d(\gamma^2+\beta^2) \tag{65}$$
$$+ 2\sum_{j<k} d(\gamma^2+\beta^2)\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})\Big)$$

$$= \frac{d(\gamma^2+\beta^2)}{n_i}\Big(1 \tag{66}$$
$$+ (n_i-1)\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})]\Big).$$

Deriving the expression for the trace of normalized covariance Combining the above results:

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) = \frac{d(\gamma^2+\beta^2)-\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2} \tag{67}$$

$$= \frac{1}{\|\boldsymbol{\mu}(\boldsymbol{X}_i)\|^2}d(\gamma^2+\beta^2) - 1 \tag{68}$$

$$= \frac{d(\gamma^2+\beta^2)}{\frac{d(\gamma^2+\beta^2)}{n_i}\left(1+(n_i-1)\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})]\right)} \tag{69}$$
$$- 1$$

$$= \frac{n_i}{1+(n_i-1)\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})]} - 1 \tag{70}$$

$$= \frac{(n_i-1)(1-\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})])}{1+(n_i-1)\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j},\boldsymbol{x}_{i,k})]}. \tag{71}$$

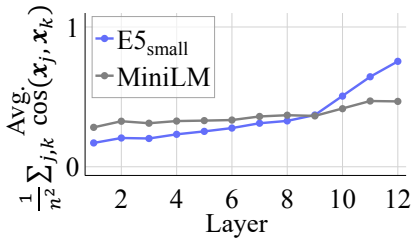


Figure 15: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j, \boldsymbol{x}_k)$ for MiniLM and E5_{small} on the Wikipedia dataset.

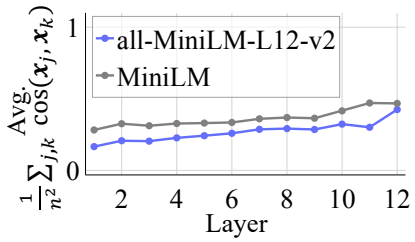


Figure 16: Layer-wise average of $\frac{1}{n^2}\sum_{j,k}\cos(\boldsymbol{x}_j, \boldsymbol{x}_k)$ for MiniLM and all-MiniLM-L12-v2 on the Wikipedia dataset.

Bounding the trace This expression is monotonically decreasing in $\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,k})]$. To verify this, we compute:

$$\frac{\partial}{\partial \mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,k})]}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) \tag{72}$$
$$= -\frac{n_i}{(1+(n_i-1)\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,k})])^2} < 0.$$

Under Assumption 2, when $\mathbb{E}_{j<k}[\cos(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,k})] = 1/3$, we have:

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) = \frac{(n_i-1)\cdot\frac{2}{3}}{1+(n_i-1)\cdot\frac{1}{3}} = \frac{2(n_i-1)}{n_i+2}. \tag{73}$$

This function is monotonically increasing in n_i:

$$\frac{\partial}{\partial n_i}\frac{2(n_i-1)}{n_i+2} = \frac{6}{(n_i+2)^2} > 0. \tag{74}$$

Furthermore, this function converges to 2 as $n_i \to \infty$:

$$\lim_{n_i\to\infty}\frac{2(n_i-1)}{n_i+2} = 2. \tag{75}$$

Therefore, $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{X}_i^{\text{norm}})) \leq 2$ holds under Assumptions 1 and 2.

K Computational Resources

All experiments in this paper were conducted using a single NVIDIA RTX 6000 Ada graphics card. In § 5, computing SOCM values for all text pairs required approximately 2 hours per model. The analyses with MTEB (eng, v2) in § 7 required approximately 6 hours per model.

L Use of AI Assistants

In preparing this paper, we utilized AI assistants (Claude, ChatGPT) to support various aspects of the writing and implementation process. These tools were employed for tasks such as code debugging, language polishing, formatting assistance, and generating visualizations. However, all research ideas, methodological designs, experimental analyses, and scientific interpretations presented in this work are entirely our own. The AI assistants served solely as technical aids and did not contribute to the conceptual or intellectual content of this research.

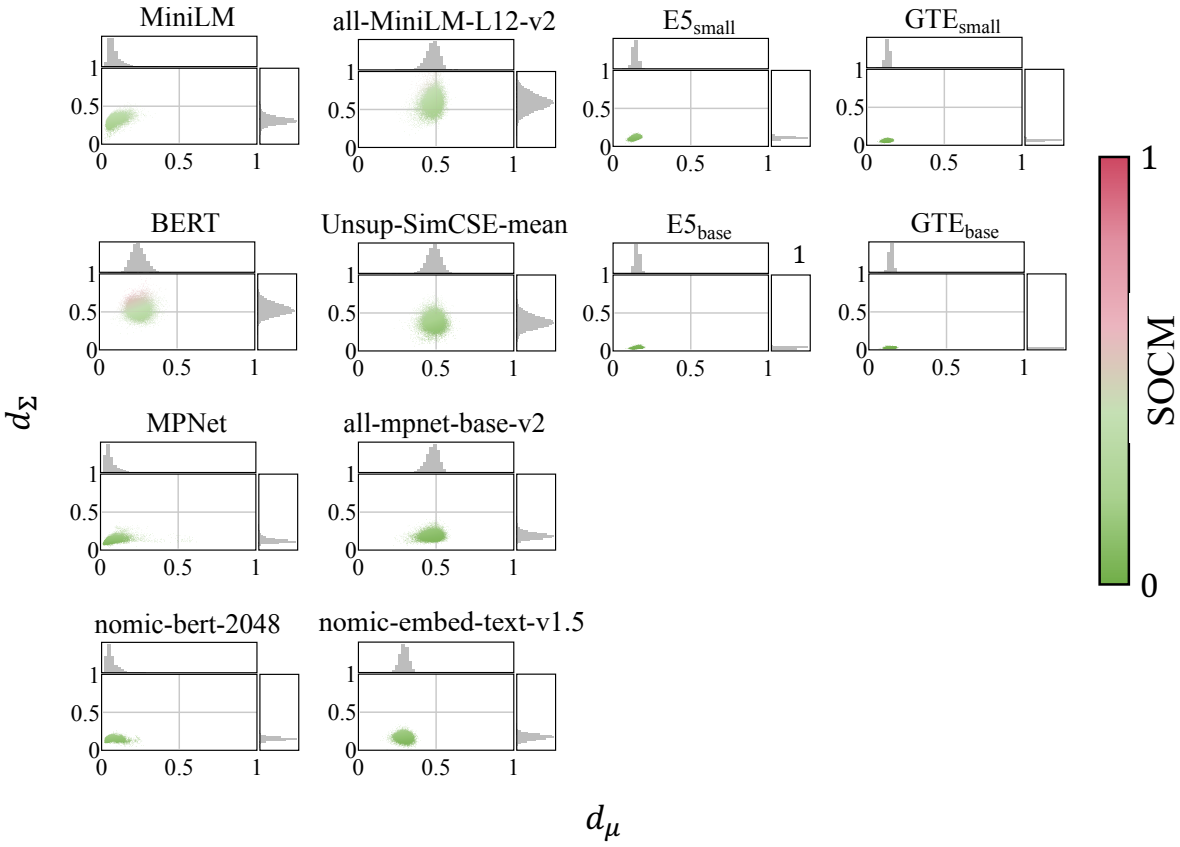


Figure 17: Scatter plots of d_μ (x) and d_Σ (y) for the models examined in § 5 on Wikipedia. Each point represents a text pair, colored by SOCM. The top and right marginal histograms show the distributions of d_μ and d_Σ.