Spread-Out Regularization in Matryoshka Text Embeddings: A Retrieval-Focused Analysis

Zeqi Zhou
Brown University
zeqi_zhou@brown.edu

Stephen Bach
Brown University
sbach@cs.brown.edu

Abstract

Recent evidence suggests that modern text embedding models contain substantial dimensional redundancy in their learned representations. This paper presents a controlled analysis of whether spread-out regularization, a batch-level loss that penalizes high pairwise similarity within a batch, can improve representation-space utilization in Matryoshka Representation Learning settings. Our results show that spread-out loss is a retrieval-biased regularizer: under MRL, it improves retrieval performance by up to +2.3 points at full dimension, while producing much smaller changes in other tasks and slightly degrading pair-classification performance. In the MRL setting, effective spread-out variants improve retrieval across all prefix dimensions, with the largest relative gains appearing at smaller dimensions where representational capacity is most constrained. The effect depends on how regularization strength is allocated across dimensions: equal weighting and smaller-dimension emphasis produce stronger retrieval performance, whereas larger-dimension emphasis underperforms, indicating that lower cosine concentration alone is not sufficient for better retrieval. A subsequent query-length analysis further shows that these gains are concentrated on short queries.

1 Introduction

Text embedding models map natural language to dense vector representations, enabling a wide range of applications including information retrieval, semantic similarity, classification, and clustering (Muennighoff et al., 2023). In dense retrieval, queries and documents are encoded independently into a shared vector space, allowing relevant documents to be identified through nearest-neighbor search (Karpukhin et al., 2020; Xiong et al., 2020). Recent models target general-purpose performance across these tasks, combining contrastive objectives with hard negatives and multi-stage training to achieve strong results on benchmarks such as MTEB (Muennighoff et al., 2023; Li et al., 2023; Lee et al., 2025).

As these models grow more capable, a natural question arises: how effectively do they use their available representation space? Recent evidence points to substantial dimensional redundancy. Takeshita et al. (2025) show that randomly removing up to 50% of embedding dimensions causes only minor degradation in retrieval and classification, and Tsukagoshi and Sasano (2025) find that prompt-based embeddings often remain effective after aggressive dimensionality reduction. These observations suggest that current models may underutilize parts of their representation space.

The question is particularly relevant for Matryoshka Representation Learning (MRL; Kusupati et al., 2022), which trains a single embedding so that prefix sub-vectors of varying dimensionality can be used independently. MRL has been adopted in recent models including Qwen3 Embedding, Jina Embeddings v3, and EmbeddingGemma (Zhang et al., 2025; Sturua et al., 2025; Vera et al., 2025). If different prefixes must all support retrieval, then the geometry and utilization of the embedding space become central to both accuracy and deployment efficiency.

One line of response is explicit geometric regularization. EmbeddingGemma incorporates a spread-out loss, building on the regularizer of Zhang et al. (2017), to improve embedding quality and space utilization (Vera et al., 2025). Given the dimensional redundancy documented above, spread-out regularization is a natural candidate for addressing underutilization, yet its effect on downstream retrieval has not been systematically studied.

In this paper, we provide a controlled analysis of spread-out regularization in text embedding training. Rather than proposing a new method, we isolate the effect of this loss across task types, embed-

ding dimensions, and query characteristics. Our analysis yields three findings:

1. **Spread-out is a retrieval-biased regularizer.** It improves retrieval by up to +2.3 points at full dimension while slightly hurting pair classification, reshaping embedding geometry in ways that favor nearest-neighbor search but not all downstream tasks.

2. **Under MRL, weighting strategy matters but more is not better.** Equal and smaller-dimension-weighted spread-out are effective, while larger-dimension weighting consistently underperforms. Better geometry does not guarantee better retrieval, revealing an alignment-uniformity trade-off.

3. **The benefit concentrates on short-query retrieval.** Spread-out helps most when queries are short, compensating for query-side geometric crowding rather than improving retrieval uniformly.

We additionally observe that the spread-out advantage is largest early in training and diminishes as contrastive learning converges.

2 Background and Related Work

2.1 Dense Text Embeddings

Dense retrieval encodes queries and passages independently into a shared vector space for efficient nearest-neighbor search (Karpukhin et al., 2020; Xiong et al., 2020). Training relies on contrastive objectives with in-batch or mined hard negatives, and models are evaluated on standardized benchmarks, which cover retrieval, semantic textual similarity, classification, and clustering. Recent general-purpose models combine retrieval supervision with broader objectives through multi-stage pipelines, achieving strong performance across diverse tasks (Li et al., 2023; Lee et al., 2025).

2.2 Embedding Geometry and Space Utilization

The quality of dense embeddings depends not only on local query–document alignment but also on the global geometry of the representation space. When embeddings concentrate in narrow regions of the hypersphere or occupy a limited subspace, similarity scores lose discriminative power and representational capacity goes unused. Wang and Isola (2020) formalize two desirable geometric properties—alignment of positive pairs and uniformity on the hypersphere—while Jing et al. (2022) identify dimensional collapse, where embeddings occupy a low-dimensional subspace, as a failure mode of contrastive learning. In the context of sentence representations, anisotropy has been shown to degrade representation quality, motivating corrections such as whitening (Su et al., 2021).

More direct evidence of dimensional redundancy comes from recent text-embedding studies. Takeshita et al. (2025) observe that randomly removing up to 50% of embedding dimensions causes only minor degradation in retrieval and classification, and representations obtained from state-of-the-art text embedders can contain a significant number of dimensions that have a negative impact on many downstream tasks. Similarly, Tsukagoshi and Sasano (2025) further show that prompt-based embeddings often remain effective after aggressive dimensionality reduction, and analyze this behavior through redundancy, isotropy, and intrinsic dimensionality. Their work has shown that the performance of text embedding models on many tasks remains virtually undiminished even if only the first 25% of the embedding dimensions are retained. These findings suggest that current models underutilize parts of their representation space, and that explicit geometric regularization may improve how that space is used.

2.3 Matryoshka Representation Learning

Matryoshka Representation Learning (MRL) trains a single embedding so that prefixes of the full vector are independently usable (Kusupati et al., 2022). Given full dimension D, MRL defines prefix dimensions $\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$ with $m_1 < m_2 < \cdots < m_K = D$. Each prefix is obtained by taking the first m coordinates and L2-normalizing independently, so that a single model supports multiple deployment regimes: shorter prefixes for low-latency retrieval and longer prefixes for higher accuracy.

3 Experimental Setup

3.1 Training

We use ModernBERT-base (Warner et al., 2025) (149M parameters, $D = 768$) as the primary backbone, fine-tuned on RLHN-680K. RLHN-680K is a hard-negative retrieval training set containing query, positive passage, and mined hard-negative passage tuples. Each training example contains 1

Hyperparameter	Value
Learning rate	1×10^{-5}
Epochs	3
Num. GPUs	8
Per-GPU batch size	16
Max sequence length	512
Temperature τ	1.0

Table 1: Training hyperparameters for the main experiment.

positive and 7 hard negatives per query ($n = 7$, $\tau = 1$). Table 1 summarizes the training hyperparameters.

3.2 Training Objective

Input. Each training example consists of a query q_i, a positive passage p_i^+, and n hard negatives $p_{i,1}^-, p_{i,2}^-, \ldots, p_{i,n}^-$. All texts are encoded into L2-normalized embeddings, so that $s(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ equals cosine similarity.

Objective. The training loss combines two terms: a contrastive loss for query–passage alignment and a spread-out regularizer for geometric dispersion.

Given a query q_i, its corresponding positive p_i^+, and n hard negatives, the contrastive loss is:

$$\ell_{\text{hard}} = -\log \frac{e^{s(\mathbf{q}_i, \mathbf{p}_i^+)/\tau}}{e^{s(\mathbf{q}_i, \mathbf{p}_i^+)/\tau} + \sum_{j=1}^{n} e^{s(\mathbf{q}_i, \mathbf{p}_{i,j}^-)/\tau}} \tag{1}$$

where τ is the temperature and $s(\cdot, \cdot)$ is cosine similarity. The batch loss $\mathcal{L}_{\text{hard}}$ averages ℓ_{hard} over all examples. To isolate the effect of spread-out regularization, in-batch negatives loss is not used in our experiments.

The second loss is based on the global orthogonal regularizer (GOR) introduced in Zhang et al. (2017) and adopted in EmbeddingGemma (Vera et al., 2025). This loss encourages embeddings to be spread out over the embedding space, to fully utilize its expressive capacity. Given a batch $\mathcal{B}$ of size B, the spread-out loss is:

$$\mathcal{L}_{\text{so}} = \frac{1}{B(B-1)} \left(\sum_{\substack{i,j \in \mathcal{B} \\ i \neq j}} s(\mathbf{q}_i, \mathbf{q}_j)^2 \right) + \frac{1}{B(B-1)} \left(\sum_{\substack{i,j \in \mathcal{B} \\ i \neq j}} s(\mathbf{p}_i^+, \mathbf{p}_j^+)^2 \right) \tag{2}$$

Minimizing $\mathcal{L}_{\text{so}}$ penalizes high pairwise cosine similarity among embeddings, reducing the off-diagonal energy of the batch Gram matrix.

MRL integration. Under MRL, both losses are computed at each prefix dimension $m \in \mathcal{M}$ using the first m coordinates of each embedding, L2-normalized independently. The combined objective averages over all prefix dimensions:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(\mathcal{L}_{\text{hard}}^{(m)} + \lambda_m \cdot \mathcal{L}_{\text{so}}^{(m)} \right) \tag{3}$$

where λ_m controls the spread-out strength at dimension m. When all $\lambda_m = 0$, this reduces to standard MRL training (HARDONLY). For non-MRL experiments, the loss is computed at the full dimension only: $\mathcal{L} = \mathcal{L}_{\text{hard}}^{(D)} + \lambda \cdot \mathcal{L}_{\text{so}}^{(D)}$. In our setting, the full embedding dimension is 768 for ModernBERT-base, and we use prefix dimensions $\mathcal{M} = \{64, 128, 256, 512, 768\}$.

3.3 Spread-Out Variants

The per-dimension spread-out weight in Equation 3 follows the parameterization $\lambda_m = \lambda_{\text{full}} \cdot (m_K/m)^a$, where $m_K = D$ is the full dimension. The exponent a controls how spread-out strength is distributed across MRL dimensions: $a > 0$ assigns larger weights to smaller dimensions, $a < 0$ to larger dimensions, and $a = 0$ applies equal weight everywhere.

The settings of four variants are compared in Table 3.

3.4 Baseline Experiment

To establish the basic effect of spread-out without MRL, we also train ModernBERT-base on RLHN-680K at the full 768 dimension only (HARDONLY vs. spread-out, no MRL prefix training).

3.5 Evaluation

The downstream performance is evaluated on 39 MTEB tasks (Muennighoff et al., 2023) spanning 6 task types: Classification (8), Clustering (8), PairClassification (3), Retrieval (10), STS (9), and Summarization (1). For MRL runs, we report performance at each prefix dimension and track all three epoch checkpoints. Our analysis focuses on the Retrieval tasks; the complete per-type breakdown across all dimensions is reported in Table 4.

3.6 Geometry Diagnostics

The embedding geometry is measured with two metrics: (1) **mean off-diagonal cosine** ($\overline{\cos}_{\text{off}}$),

Variant	Dim	Short <100 Score	Δ%	Medium 100–300 Score	Δ%	Long >300 Score	Δ%	Overall Avg Score	Δ%
HARDONLY	64	9.12	–	**49.56**	–	**50.57**	–	15.44	–
	128	15.76	–	**59.12**	–	57.68	–	22.44	–
	256	24.89	–	**62.94**	–	**61.16**	–	30.73	–
	512	29.24	–	**65.89**	–	**62.48**	–	34.80	–
	768	31.02	–	**66.81**	–	**63.51**	–	36.45	–
Unweighted	64	**11.29**	+23.8%	47.96	−3.2%	49.93	−1.3%	**17.06**	+10.5%
	128	16.89	+7.2%	57.88	−2.1%	56.71	−1.7%	23.21	+3.4%
	256	24.96	+0.3%	62.45	−0.8%	60.43	−1.2%	30.70	−0.1%
	512	30.91	+5.7%	64.76	−1.7%	61.86	−1.0%	36.05	+3.6%
	768	**33.28**	+7.3%	65.53	−1.9%	62.31	−1.9%	**38.16**	+4.7%
Weighted	64	10.35	+13.5%	48.62	−1.9%	49.51	−2.1%	16.32	+5.8%
	128	**17.61**	+11.7%	57.84	−2.2%	**58.29**	+1.1%	**23.87**	+6.4%
	256	**25.75**	+3.5%	62.33	−1.0%	59.87	−2.1%	**31.33**	+2.0%
	512	**31.08**	+6.3%	64.79	−1.7%	61.03	−2.3%	**36.17**	+3.9%
	768	32.67	+5.3%	66.05	−1.1%	62.04	−2.3%	37.70	+3.4%
Weighted-Incr	64	8.74	−4.2%	46.41	−6.4%	49.84	−1.4%	14.72	−4.6%
	128	15.20	−3.6%	56.87	−3.8%	57.14	−0.9%	21.68	−3.4%
	256	23.93	−3.9%	61.82	−1.8%	61.33	+0.3%	29.79	−3.0%
	512	30.07	+2.8%	64.56	−2.0%	62.12	−0.6%	35.33	+1.5%
	768	31.60	+1.9%	65.44	−2.1%	63.00	−0.8%	36.76	+0.9%
Baseline (no MRL, dim 768)									
HARDONLY	768	31.35	–	59.48	–	56.68	–	35.61	–
SPREADOUT	768	33.60	+7.2%	58.76	−1.2%	56.76	+0.1%	37.43	+5.1%

Table 2: Retrieval scores by query length bin across MRL dimensions (final epoch), with Δ% relative to HARDONLY at each dimension.

Variant	SO	λ_{full}	a
HARDONLY	No	0.0	0.0
Unweighted	Yes	1.0	0.0
Weighted	Yes	1.5	0.5
Weighted-Incr	Yes	1.5	−0.5

Table 3: Spread-out (SO) loss variants. The per-dimension weight is $\lambda_m = \lambda_{\text{full}} \cdot (m_K/m)^a$.

the average absolute cosine similarity between all pairs of distinct embeddings, where lower values indicate less directional concentration; and (2) s_2, the squared ratio of the second to the first singular value of the embedding matrix, where lower values indicate less dominance by a single direction. Both metrics are computed on retrieval-task test samples with up to 2000 pairs per query-length bin, at all five MRL dimensions, separately for query and passage sides.

4 Results

4.1 RQ1: Does Spread-Out Improve Dense Retrieval?

Starting with the simplest setting: a single embedding dimension with no MRL. In the no-MRL baseline (Table 7, bottom rows), adding spread-out regularization improves retrieval from 31.40 to 34.04 at epoch 3, a gain of +2.6 points. At the same checkpoints, the overall 39-task average improves by only +1.6, making the retrieval gain roughly 2–3× the overall gain. Spread-out is not a uniform performance booster—it disproportionately benefits retrieval.

The per-type breakdown in Table 4 confirms this asymmetry under MRL. At dimension 768, the Unweighted variant improves Retrieval by +2.3 over HARDONLY while PairClassification drops by −1.3. Classification, Clustering, and STS show small positive changes, and Summarization is mixed across variants. This pattern holds across dimensions: spread-out reshapes embedding geometry in ways that benefit nearest-neighbor ranking—the core operation in retrieval—at some cost to tasks that depend on fine-grained pairwise similarity judgments.

4.2 RQ2: How Does Spread-Out Interact with MRL?

Having established that spread-out is a retrieval-biased regularizer, we now ask how it interacts with MRL, where the model must serve multiple prefix dimensions simultaneously. The Retrieval

Variant	Dim	Cls	Clu	PairCls	Retrieval	STS	Summ	Avg
HARDONLY	64	54.58	38.46	70.56	10.08	64.73	8.45	41.14
	128	59.49	40.33	72.25	16.54	65.56	11.72	44.31
	256	62.60	41.66	73.15	25.10	67.23	14.42	47.36
	512	64.35	42.09	73.37	29.85	67.92	14.49	48.68
	768	64.88	42.45	**73.46**	31.55	68.45	13.84	49.11
Unweighted	64	55.51	39.02	70.19	12.68	66.36	9.87	42.27
	128	59.91	40.54	71.42	18.05	66.11	10.78	44.47
	256	62.86	41.74	71.57	25.53	67.65	11.81	46.86
	512	64.73	42.23	72.00	31.80	68.60	13.53	48.82
	768	65.14	42.58	72.12	**33.81**	**69.11**	13.70	49.41
Weighted	64	54.50	38.88	69.19	11.12	64.60	11.36	41.61
	128	59.68	40.38	70.81	18.36	65.67	13.79	44.78
	256	62.80	41.72	71.46	26.59	67.08	15.72	47.56
	512	64.69	42.07	71.78	31.85	67.91	16.67	49.16
	768	**65.17**	42.55	71.96	33.66	68.30	**16.00**	**49.61**
Weighted-Incr	64	54.18	38.82	70.15	10.19	64.26	6.28	40.65
	128	59.26	40.78	71.37	16.69	64.97	11.42	44.08
	256	62.43	41.87	72.19	24.92	66.67	15.17	47.21
	512	64.39	42.67	72.55	30.86	67.43	15.20	48.85
	768	65.02	**42.81**	72.67	32.51	67.86	14.50	49.23

Table 4: Per-task-type MTEB scores across all MRL dimensions at the final epoch. Cls = Classification (8 tasks), Clu = Clustering (8), PairCls = PairClassification (3), STS = Semantic Textual Similarity (9), Summ = Summarization (1), Avg = mean of 6 type averages. Bold marks the best score per task type.

column of Table 4 shows that Unweighted (33.81) and Weighted (33.66) both substantially outperform HARDONLY (31.55) at the full 768 dimension, while Weighted-Incr gains only +1.0. The same ranking holds at every prefix dimension, establishing that equal or smaller-dimension weighting is consistently stronger, while larger-dimension weighting is substantially weaker and less stable.

The dimension-level pattern reveals where spread-out helps most. Unweighted shows its largest absolute gain at dim 64 (+2.6), where geometric capacity is most constrained, while its gain at dim 768 is +2.3. Weighted distributes gains more evenly across dimensions. This suggests that smaller prefix spaces, which must pack all discriminative information into fewer coordinates, benefit more from the dispersion that spread-out provides.

An interesting discrepancy emerges between geometry and performance. The geometry diagnostics in Tables 5–6 show that Weighted produces the most dispersed embeddings overall (query-side $\overline{\cos}$ 0.278 vs. Unweighted 0.280 at dim 768), yet Unweighted achieves a slightly higher retrieval score (33.81 vs. 33.66). This points to an alignment-uniformity trade-off: retrieval requires both geometric dispersion and accurate query-passage alignment, and excessive regularization toward uniformity can compromise the ranking signal.

4.3 RQ3: Does the Benefit Vary Across Retrieval Conditions?

The aggregate gains reported in Sections 4.1–4.2 turn out to be driven almost entirely by short queries. Table 2 breaks down retrieval performance by query length across all MRL dimensions. Short queries (<100 tokens) make up 84.5% of the retrieval workload, and this is precisely where spread-out helps: at dimension 768, Unweighted improves short-query retrieval by +7.3% over HARDONLY (33.28 vs. 31.02), driving a +4.7% gain in the overall main score. For medium and long queries, however, spread-out is much less consistently beneficial: HARDONLY is often the strongest variant, and most spread-out variants show small negative deltas, with only a few exceptions.

The embedding geometry explains why. Tables 5–6 show that spread-out primarily reshapes query-side representations, and the effect is strongly query-length-dependent. For short queries, the effective spread-out variants, especially Unweighted and Weighted, generally reduce directional concentration compared to HARDONLY. (e.g., Unweighted overall $\overline{\cos}$ 0.280 vs. HARDONLY 0.301 at dim 768), but for medium and long queries, Unweighted and Weighted-Incr are actually *more* concentrated than HARDONLY. Weighted shows the most consistent dispersion pattern across query lengths, especially at higher di-

Variant	Dim	Short <100		Med 100–300		Long >300		Overall	
		$\overline{\cos}$	s_2	$\overline{\cos}$	s_2	$\overline{\cos}$	s_2	$\overline{\cos}$	s_2
HARDONLY	64	0.384	0.176	0.160	0.047	0.169	0.045	0.359	0.162
	128	0.373	0.162	0.162	0.040	0.160	0.036	0.350	0.148
	256	0.324	0.127	0.162	0.037	0.177	0.041	0.306	0.117
	512	0.322	0.125	0.157	0.034	0.166	0.037	0.304	0.115
	768	0.318	0.123	0.161	0.034	0.165	0.036	0.301	0.113
Unweighted	64	0.304	0.123	0.181	0.054	0.183	0.050	0.291	0.115
	128	0.362	0.152	0.190	0.050	0.183	0.046	0.343	0.141
	256	0.325	0.127	0.175	0.041	0.187	0.046	0.308	0.117
	512	0.303	0.112	0.168	0.037	0.182	0.042	0.289	0.104
	768	0.293	0.107	0.174	0.039	0.182	0.042	0.280	0.099
Weighted	64	0.348	0.150	0.143	0.042	0.184	0.050	0.327	0.139
	128	0.339	0.137	0.162	0.040	0.182	0.044	0.320	0.126
	256	0.306	0.116	0.150	0.033	0.172	0.040	0.290	0.107
	512	0.299	0.109	0.142	0.029	0.158	0.034	0.282	0.101
	768	0.295	0.107	0.142	0.029	0.156	0.034	0.278	0.099
Weighted-Incr	64	0.375	0.167	0.218	0.067	0.228	0.068	0.358	0.156
	128	0.373	0.159	0.207	0.056	0.211	0.056	0.355	0.148
	256	0.330	0.129	0.191	0.046	0.198	0.049	0.315	0.120
	512	0.316	0.120	0.181	0.042	0.187	0.044	0.301	0.111
	768	0.311	0.117	0.187	0.044	0.188	0.044	0.298	0.109

Table 5: Query-side embedding geometry across all MRL dimensions, measured on retrieval-task test samples and binned by query length. $\overline{\cos}$ = mean off-diagonal cosine similarity; s_2 = squared ratio of second to first singular value. Lower values indicate more dispersed embeddings.

mensions. The passage side, in contrast, is barely affected: geometry differences are an order of magnitude smaller ($\Delta\overline{\cos} \approx 0.002$ on passages vs. 0.020 on queries at dim 768). Spread-out thus compensates for the geometric crowding that occurs when short, ambiguous queries are mapped to a shared embedding space. When queries are already long and semantically rich, the contrastive hard-negative signal alone provides sufficient discrimination, and the additional dispersion may slightly interfere with ranking.

4.4 Training Dynamics

Table 7 tracks retrieval at all three epoch checkpoints. The spread-out advantage is largest at epoch 1 and diminishes over training: at dimension 768, the Unweighted gain over HARDONLY drops from +4.4 (epoch 1) to +2.5 (epoch 2) to +2.3 (epoch 3). The no-MRL baseline shows the same pattern even more starkly, with the epoch 1 gain (+7.4) nearly three times the epoch 3 gain (+2.6).

The temporal pattern suggests that spread-out helps organize the retrieval space before hard-negative contrastive learning has converged. Once the contrastive objective has shaped the embedding geometry sufficiently, the marginal contribution of regularization diminishes. Weighted-Incr is an exception to this smooth decay: it peaks at epoch 2 (+3.2 at dim 768) and drops at epoch 3 (+1.0), indicating less stable training dynamics than the other variants.

5 Discussion

Selective rather than universal benefit. The query-length breakdown shows that spread-out is conditionally useful rather than uniformly beneficial: at dimension 768, Unweighted improves short-query retrieval by +7.3% but slightly hurts medium- and long-query retrieval. This suggests that spread-out compensates for geometric crowding when the query signal is short or under-specified.

Toward query-adaptive regularization. The concentration of gains on short queries suggests that a fixed, query-length-agnostic spread-out weight may be suboptimal. A natural extension is to make the regularization strength query-adaptive: increasing it for short queries and reducing it for longer queries. For example, one simple design would assign spread-out loss weights inversely proportional to the logarithm of query length.

$$\lambda_i^{(q)} \propto \frac{1}{\log(L_i + 1)},$$

where L_i is the length of the i-th query. This would redistribute spread-out pressure toward short

Variant	Dim	Short <100		Med 100–300		Long >300		Overall	
		$\overline{\cos}$	s_2	$\overline{\cos}$	s_2	$\overline{\cos}$	s_2	$\overline{\cos}$	s_2
HARDONLY	64	0.040	0.023	0.114	0.034	0.159	0.047	0.050	0.025
	128	0.042	0.016	0.114	0.028	0.129	0.030	0.050	0.017
	256	0.037	0.012	0.125	0.027	0.135	0.030	0.047	0.014
	512	0.036	0.010	0.118	0.023	0.119	0.024	0.045	0.012
	768	0.035	0.010	0.118	0.022	0.116	0.022	0.044	0.011
Unweighted	64	0.042	0.023	0.118	0.036	0.149	0.041	0.051	0.025
	128	0.040	0.016	0.115	0.028	0.119	0.027	0.049	0.018
	256	0.035	0.012	0.117	0.024	0.123	0.026	0.044	0.013
	512	0.034	0.010	0.113	0.022	0.117	0.023	0.043	0.011
	768	0.034	0.009	0.113	0.021	0.111	0.021	0.042	0.011
Weighted	64	0.036	0.023	0.110	0.034	0.161	0.050	0.045	0.024
	128	0.037	0.016	0.115	0.028	0.138	0.032	0.046	0.017
	256	0.035	0.012	0.115	0.024	0.129	0.028	0.044	0.013
	512	0.034	0.010	0.114	0.022	0.115	0.022	0.043	0.011
	768	0.033	0.009	0.113	0.021	0.112	0.021	0.042	0.011
Weighted-Incr	64	0.042	0.023	0.130	0.038	0.150	0.043	0.052	0.025
	128	0.038	0.016	0.118	0.028	0.119	0.029	0.046	0.017
	256	0.037	0.012	0.122	0.026	0.124	0.027	0.046	0.014
	512	0.034	0.010	0.121	0.023	0.118	0.023	0.044	0.012
	768	0.034	0.010	0.119	0.023	0.113	0.021	0.043	0.011

Table 6: Passage-side embedding geometry across all MRL dimensions, measured on retrieval-task test samples and binned by query length. $\overline{\cos}$ = mean off-diagonal cosine similarity; s_2 = squared ratio of second to first singular value. Lower values indicate more dispersed embeddings.

queries while reducing unnecessary regularization on longer queries.

Asymmetric effects on query and passage geometry. The geometry diagnostics reveal that spread-out does not affect query and passage representations symmetrically. Query embeddings remain far more directionally concentrated than passage embeddings: at dimension 768 under HARDONLY, the overall query-side $\overline{\cos}$ is 0.301, compared with only 0.044 on the passage side. The passage side also responds more predictably to weighting changes—moving from Unweighted to Weighted reduces short-query passage-side $\overline{\cos}$ from 0.042 to 0.036 at dimension 64—whereas the same change on the query side is less stable, increasing $\overline{\cos}$ at some dimensions while reducing it at others. A likely explanation is that queries within the same dataset often share surface format or task-specific templates, creating a strong "query-type" direction that a batch-level loss struggles to disperse without disrupting useful query–passage alignment.

Separating query-side and passage-side effects. These asymmetries suggest that treating spread-out as a single global constraint is too coarse. A direct ablation—training with query-only spread-out, passage-only spread-out, and the current two-sided formulation—would clarify whether retrieval gains are driven by reducing query-side crowding, improving passage-side coverage, or the interaction between the two. It would also reveal whether the instability on the query side is inherent to regularizing short, template-like queries, or an artifact of using the same strength for both sides.

Dispersion versus alignment. More broadly, our findings indicate that the goal is not to make embeddings as dispersed as possible. The best-performing variant is not always the one with the lowest cosine concentration: retrieval requires a balance between geometric dispersion and query–passage alignment. Spread-out is most beneficial when it reduces harmful crowding in low-information regimes—especially short-query retrieval—but excessive or poorly targeted dispersion can interfere with the ranking signal. This motivates adaptive objectives that vary spread-out strength by query length, representation side, and potentially training stage.

6 Limitations

This study has several limitations.

First, all experiments use a single backbone (ModernBERT-base) and a single training dataset (RLHN-680K). Due to GPU and time constraints, we were unable to train on larger models or addi-

Variant	Dim	Ep. 1	Ep. 2	Ep. 3
HARDONLY	64	7.88	9.95	10.08
	128	12.11	16.41	16.54
	256	18.69	23.03	25.10
	512	23.34	27.09	29.85
	768	24.37	28.35	31.55
Unweighted	64	9.73	11.77	**12.68**
	128	14.99	16.71	18.05
	256	21.97	22.42	25.53
	512	27.17	28.48	31.80
	768	28.80	30.80	**33.81**
Weighted	64	9.32	9.79	11.12
	128	13.68	16.73	**18.36**
	256	21.10	23.39	**26.59**
	512	25.92	28.20	**31.85**
	768	27.36	29.56	33.66
Weighted-Incr	64	8.59	10.97	10.19
	128	12.33	17.65	16.69
	256	18.44	24.62	24.92
	512	23.95	30.42	30.86
	768	25.44	31.51	32.51
Baseline (no MRL, dim 768)				
HARDONLY	768	30.48	29.04	31.40
SPREADOUT	768	37.89	33.16	34.04

Table 7: Training dynamics: Retrieval scores (10-task nDCG@10) across training epochs and MRL dimensions. Baseline rows are no-MRL runs. Bold marks the best score per prefix dimension.

tional architectures. Testing whether spread-out generalizes across model scales, decoder-based architectures, and different training data distributions is an important direction for future work.

Second, spread-out loss is batch-based. Its behavior may depend on batch size, the number of hard negatives, and the composition of training batches. We do not vary these factors systematically.

Third, our geometry diagnostics are computed on retrieval-task test samples with up to 2000 pairs per bin. This is sufficient for directionality analysis but is not a full multi-task geometry aggregate.

Finally, this work analyzes an existing regularizer rather than proposing a new method. Its contribution is diagnostic: it clarifies when and how spread-out helps, rather than advancing the state of the art in retrieval performance.

$$L = -\log \frac{e^{s_{pos}/\tau}}{\sum_{j=0}^{G-1} e^{s_j/\tau}}$$

7 Conclusion

We presented a controlled empirical analysis of spread-out regularization in Matryoshka text embedding training, organized around three research questions.

Spread-out acts as a retrieval-biased geometry regularizer (RQ1), improving retrieval by up to +2.3 points under MRL while slightly hurting pair classification. Under MRL, equal and smaller-dimension-weighted spread-out are effective, while larger-dimension weighting is consistently weaker, revealing an alignment-uniformity trade-off (RQ2). The retrieval benefit concentrates on short-query examples, which make up the dominant retrieval workload, and does not help when queries are already semantically rich (RQ3).

More broadly, the consistent retrieval gains across settings, particularly short queries, which dominate the evaluated retrieval workload, indicate that spread-out regularization has substantial potential as a training component for embedding models, especially retrieval-oriented systems. Equal spread-out weighting provides a simple and effective default, and we expect further gains from adaptive weighting strategies and integration with richer training recipes.

References

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2022. Understanding dimensional collapse in contrastive self-supervised learning. *Preprint*, arXiv:2110.09348.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

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

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

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.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding

benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2025. Jina embeddings v3: Multilingual text encoder with low-rank adaptations. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V*, page 123–129, Berlin, Heidelberg. Springer-Verlag.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *Preprint*, arXiv:2103.15316.

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

Hayato Tsukagoshi and Ryohei Sasano. 2025. Redundancy, isotropy, and intrinsic dimensionality of prompt-based text embeddings. *Preprint*, arXiv:2506.01435.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *Preprint*, arXiv:2509.20354.

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'20. JMLR.org.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. 2017. Learning spread-out local feature descriptors. *Preprint*, arXiv:1708.06320.

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