When Queries Crowd: Diagnosing and Improving Spread-Out Regularization for Text Embeddings

Anonymous ACL submission

Abstract

Text embedding models are central to information retrieval and deep research tasks, and their behaviors are strongly shaped by the training objective. While recent systems have largely converged on InfoNCE-style loss functions, there have been continued attempts to design additional loss functions beyond this core objective. One such component is spread-out loss, a batch-level penalty on high pairwise similarity that encourages embeddings to occupy the representation space more evenly. However, in existing systems, this design is usually combined with many other design choices and has not been sufficiently ablated. To isolate its effect, we use spread-out loss as a controlled lens for studying geometric regularization in text embedding training. Across 13 retrieval benchmarks, spread-out improves fine-tuned retrievers over contrastive baselines across architectures and model sizes. We then diagnose where these gains arise and find that they concentrate on short queries, where spread-out reduces query-side directional crowding. Based on this diagnosis, we propose QUARTO, a **Qu**ery-length **A**daptive sp**r**ead-ou**t** l**o**ss that applies stronger dispersion pressure to shorter queries and weaker pressure to longer queries. QUARTO further improves retrieval performance over fixed-weight spread-out. Overall, our findings point to the potential of embedding-space regularization and suggest that text-embedding training should explore a broader design space of objectives beyond InfoNCE-style contrastive learning.

1 Introduction

Text embedding models map natural language to dense vector representations where semantically similar texts are placed nearby in the embedding space, making them widely used for information retrieval, semantic textual similarity, and clustering tasks (Muennighoff et al., 2023). The properties of these representations are therefore not determined by architecture alone, but by the training objectives that define which distinctions the embedding space is encouraged to preserve. Modern text embedding systems have largely converged on InfoNCE-style (van den Oord et al., 2019) contrastive objectives with mined hard negatives, often combined with large-scale data and multi-stage training pipelines (Li et al., 2023; Lee et al., 2025; Zhang et al., 2025). Beyond optimizing pairwise discrimination between positives and negatives, these objectives also shape the global geometry of the embedding space. Prior analyses show that contrastive learning encourages positive-pair alignment and hyperspherical uniformity, yet its repulsive pressure does not necessarily prevent representations from concentrating in lower-dimensional subspaces (Wang and Isola, 2020; Jing et al., 2022).

A related pattern appears in recent studies on dimensional redundancy in text embeddings. These studies find that not all embedding dimensions contribute equally to downstream performance: discarding or compressing a large fraction of dimensions often causes only minor degradation (Takeshita et al., 2025; Tsukagoshi and Sasano, 2025). If large portions of the representation can be removed with limited performance loss, current text embedding models may not be fully utilizing the available embedding space. This makes explicit geometric regularization a natural direction to examine. Spread-out loss instantiates this idea by penalizing high pairwise similarity within a batch, encouraging embeddings to occupy more directions in the representation space rather than crowding around a small set of shared directions (Zhang et al., 2017). Recent text embedding systems have incorporated spread-out loss into their training objectives (Vera et al., 2025; Akram et al., 2026). However, because these systems introduce spread-out loss together with other training components, its isolated contribution remains difficult to assess. As a result, it remains unclear whether the

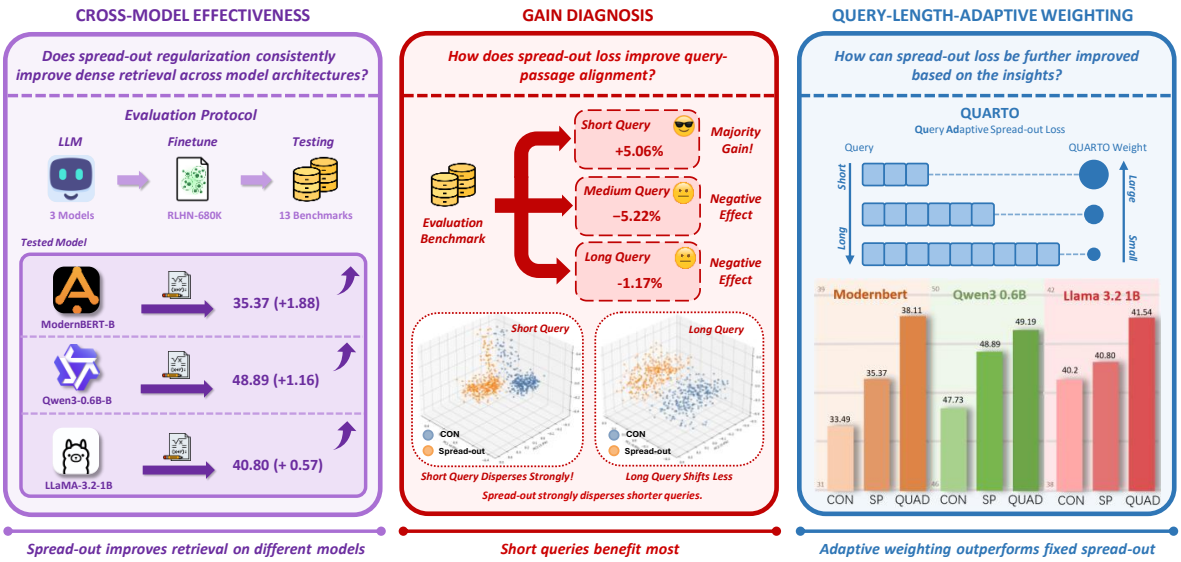

Figure 1: Overview of this work. **Left**: We train three model architectures with and without spread-out regularization on RLHN-680K and evaluate on 13 retrieval benchmarks, showing that spread-out improves retrieval in our evaluated settings. **Middle**: We analyze where the gain originates by breaking down performance by query length and examining embedding geometry, finding that short queries account for nearly all improvement; CON denotes contrastive training and SP denotes spread-out. **Right**: The diagnosis motivates QUARTO, a query-length adaptive weighting of spread-out regularization that modulates regularization strength by query length and outperforms fixed-weight spread-out.

effects of spread-out regularization persist across architectures and settings, or what representation-level changes account for any observed gains.

To bridge this gap, we treat spread-out regularization as an object of controlled study rather than as one component in a larger embedding recipe. Our analysis follows a simple chain: isolate the regularizer under a fixed hard-negative contrastive training pipeline, diagnose the query conditions and representation-level changes under which it helps, and use that diagnosis to derive an adaptive variant. We evaluate this chain across multiple model architectures and sizes, then move beyond aggregate retrieval scores by analyzing performance across query-length conditions and embedding-geometry diagnostics. The resulting diagnosis motivates QUARTO, a lightweight query-length-adaptive spread-out loss that varies the strength of geometric regularization according to query length. Figure 1 provides an overview of the study.

Our study makes three contributions:

1. **Controlled isolation of spread-out regularization.** We separate spread-out loss from other training components and evaluate it under a fixed hard-negative contrastive pipeline, showing that it improves retrieval over contrastive-only baselines across the architectures and model sizes considered in this work.

2. **Diagnosis of where spread-out helps.** We show that the gains are concentrated on short queries and connect this pattern to reduced query-side directional crowding in the embedding space.

3. **Diagnosis-driven adaptive regularization.** We translate this diagnosis into QUARTO, a lightweight query-length-adaptive variant that applies stronger spread-out pressure to shorter queries and weaker pressure to longer queries. Across our evaluated settings, QUARTO further improves retrieval performance over fixed-weight spread-out loss.

2 Related Work

2.1 Text Embedding Models and Training Objectives

Dense retrieval encodes queries and passages independently into a shared vector space for efficient nearest-neighbor search (Karpukhin et al., 2020; Xiong et al., 2021). Recent general-purpose text embedding systems largely retain this contrastive retrieval core while scaling it through multi-stage

training, hard-negative mining, broad task mixtures, synthetic data, and stronger backbone models. Representative examples include GTE (Li et al., 2023), NV-Embed (Lee et al., 2025), and Qwen3 Embedding (Zhang et al., 2025). Despite differences in architecture, data, and engineering choices, these systems reflect a common pattern: InfoNCE-style contrastive losses with in-batch or mined hard negatives remain the dominant training signal.

More recent compact embedding recipes also add auxiliary objectives on top of this core. EmbeddingGemma (Vera et al., 2025) and Jina Embeddings v5 (Akram et al., 2026), for example, both combine contrastive retrieval training with teacher distillation and GOR-style spread-out regularization. Originally proposed by Zhang et al. (2017) for visual feature descriptors, spread-out regularization penalizes high pairwise similarity within a batch, encouraging embeddings to occupy the representation space more evenly and improving robustness in settings such as quantization and approximate nearest-neighbor retrieval. However, because spread-out is introduced together with other recipe components such as distillation, task mixtures, model initialization, adapters, and checkpoint merging, its isolated contribution to retrieval performance remains unclear. Our controlled setting therefore isolates the contribution of spread-out regularization itself, rather than treating EmbeddingGemma or Jina v5 as direct baselines.

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 that some dimensions can even have a negative impact on downstream tasks. Similarly, Tsukagoshi and Sasano (2025) show that prompt-based embeddings often remain effective after retaining only 25% of dimensions, and analyze this behavior through redundancy, isotropy, and intrinsic dimensionality. These findings suggest that current models underutilize parts of their representation space, and provide a geometric motivation for studying regularizers like spread-out that aim to improve space utilization. However, because InfoNCE-style contrastive training already imposes negative repulsion and some implicit uniformity pressure, dimensional redundancy alone does not establish that an additional geometric regularizer will improve retrieval; our work addresses this gap by isolating spread-out on top of a fixed contrastive objective.

3 Preliminaries

3.1 Contrastive Training Objective

We first formalize the baseline objective used throughout the study, then define the spread-out penalty that is added on top of it. This setup lets us isolate the effect of geometric regularization while keeping the retrieval training pipeline fixed. Unlike the full recipes of recent embedding systems, this controlled objective does not include distillation, task-specific adapters, checkpoint merging, or other auxiliary losses. 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.

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.

3.2 Spread-Out Regularization

The spread-out term is based on the global orthogonal regularizer (GOR) introduced by Zhang et al. (2017). Given a batch $\mathcal{B}$ of size B, the spread-out

loss is:

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

Following recent text-embedding uses of GOR, we apply this penalty to query embeddings and positive-passage embeddings in the batch. This form treats the query-side and positive-passage-side penalties as two separately averaged terms that are summed. Minimizing $\mathcal{L}_{\text{so}}$ penalizes high pairwise cosine similarity among embeddings, reducing the off-diagonal energy of the batch Gram matrix and encouraging the representation space to be more fully utilized.

We compare two training conditions. The CONTRASTIVE baseline trains with the contrastive loss only ($\mathcal{L} = \mathcal{L}_{\text{hard}}$). SPREADOUT adds the GOR penalty to the contrastive loss ($\mathcal{L} = \mathcal{L}_{\text{hard}} + \mathcal{L}_{\text{so}}$).

4 Experiments

4.1 Setup

RLHN-680K (Thakur et al., 2025) serves as our fine-tuning corpus. It contains query, positive-passage, and mined hard-negative tuples from seven retrieval datasets in the BGE training collection, with most pairs drawn from MS MARCO (Nguyen et al., 2017). Its hard negatives are refined through a cascading LLM-judge pipeline using GPT-4o-mini and GPT-4o to identify and relabel false negatives. For training, each query is paired with one positive passage and seven hard negatives; additional dataset details, including source composition and query-length statistics, are reported in Appendix A. We select RLHN-680K for two reasons: its cleaned hard negatives provide a strong training signal, reducing the risk that data quality confounds our analysis, and using the same fixed training collection across objectives allows us to isolate the effect of spread-out regularization.

We evaluate on three model architectures: ModernBERT-base (Warner et al., 2025) (149M parameters, $D = 768$), Qwen3-0.6B-Base (Qwen Team, 2025) (596M, $D = 1024$), and Llama-3.2-1B (Meta AI, 2024) (1.2B, $D = 2048$), spanning encoder-only and decoder-only architectures with 149M to 1.2B parameters. This model set lets us test whether spread-out behavior is consistent across model families and scales. All models are trained with `BF16` mixed precision; detailed training configurations are reported in Appendix B.

The experimental protocol follows the same isolation principle across all comparisons. For a given backbone, the CONTRASTIVE and SPREADOUT runs use the same RLHN tuples, hard-negative structure, pooling rule, precision setting, and evaluation pipeline; the objective is the intended intervention. This design makes the resulting differences interpretable as objective-level effects rather than changes in data construction, model configuration, or benchmark selection.

Retrieval performance is evaluated on 13 benchmarks selected from MTEB (Muennighoff et al., 2023): ArguAna (Wachsmuth et al., 2018), CQADupstackGaming and CQADupstackUnix (Hoogeveen et al., 2015), ClimateFEVER (Diggelmann et al., 2021), FEVER (Thorne et al., 2018), FiQA2018 (Maia et al., 2018), HotpotQA (Yang et al., 2018), SCIDOCS (Cohan et al., 2020), TRECCOVID (Voorhees et al., 2020), Touché2020 (Bondarenko et al., 2020), LegalSummarization (Manor and Li, 2019), and AILACasedocs and AILAStatutes (Bhattacharya et al., 2019). Appendix D provides additional information about this evaluation suite, including benchmark descriptions and query-length distributions. The primary metric is nDCG@10, macro-averaged across the 13 tasks.

4.2 Results

The experiments in this section are designed to answer two research questions: (RQ1) Does spread-out regularization improve retrieval across model architectures and families? (RQ2) Does this benefit extend to Matryoshka Representation Learning across prefix dimensions?

RQ1: Cross-architecture generalization. Table 1 presents retrieval results for CONTRASTIVE and SPREADOUT across three model architectures.

At the aggregate level, SPREADOUT improves the 13-task Overall score for all three evaluated architectures. The gain is largest for ModernBERT-base (33.49 → 35.37, +1.88), followed by Qwen3-0.6B-Base (47.73 → 48.89, +1.16) and Llama-3.2-1B (40.23 → 40.80, +0.57). These results show that spread-out regularization improves retrieval in our evaluated settings across both encoder-only and decoder-only backbones.

The task-level gains are not uniform across the benchmark suite. Broadly, larger improvements appear more often on benchmarks with shorter queries, while longer-query benchmarks show smaller or less consistent changes. Thus, SPREADOUT provides consistent aggregate gains,

Model	Condition	Argu	CQA-G	CQA-U	CFEV	FEV	FiQA	HpQA	SCI	TREC	Tché	Legal	AILA-C	AILA-S	Overall
ModernBERT-base	CONTRASTIVE	56.10	43.36	25.99	18.11	48.85	21.32	46.65	9.05	49.42	34.40	49.75	14.36	18.09	33.49
	SPREADOUT	53.31	40.91	25.38	18.65	60.46	20.48	50.89	9.38	53.73	31.74	55.19	20.23	19.54	35.37
Qwen3-0.6B-Base	CONTRASTIVE	64.07	51.26	37.94	29.44	83.34	38.35	69.65	21.78	70.49	45.82	62.40	20.77	25.15	47.73
	SPREADOUT	63.30	52.88	38.85	30.01	84.56	38.14	71.42	21.60	77.45	47.47	62.81	21.67	25.45	48.89
Llama-3.2-1B	CONTRASTIVE	68.34	40.64	34.05	29.74	75.57	27.90	65.96	18.20	27.30	20.41	57.87	28.02	29.03	40.23
	SPREADOUT	67.40	42.22	33.42	27.35	77.23	30.20	67.19	19.00	26.12	26.78	59.60	26.81	27.06	40.80

Table 1: Retrieval nDCG@10 on the 13-task retrieval benchmark suite. Argu = ArguAna, CQA-G/U = CQADupstack Gaming/Unix, CFEV = ClimateFEVER, FEV = FEVER, HpQA = HotpotQA, SCI = SCIDOCS, TREC = TRECCOVID, Tché = Touché2020, Legal = LegalSummarization, AILA-C = AILACasedocs, and AILA-S = AILAStatutes. Overall denotes the macro average across all 13 tasks.

Model	Dim.	CON	SP	$\Delta\%$
ModernBERT-base	64	16.31	17.42	+6.8
	128	24.52	23.47	−4.3
	256	29.88	29.64	−0.8
	512	32.38	33.00	+1.9
	768	33.47	34.54	+3.2
Qwen3-0.6B-Base	64	26.97	26.89	−0.3
	128	38.27	38.33	+0.2
	256	43.17	43.71	+1.3
	512	46.80	47.26	+1.0
	1024	48.39	48.95	+1.2
Llama-3.2-1B	64	21.67	22.53	+4.0
	128	29.65	30.18	+1.8
	256	33.82	33.73	−0.3
	512	37.83	37.99	+0.4
	1024	40.51	40.77	+0.6
	2048	41.62	42.20	+1.4

Table 2: Retrieval nDCG@10 under MRL training by prefix dimension for ModernBERT-base, Qwen3-0.6B-Base, and Llama-3.2-1B. CON = CONTRASTIVE and SP = SPREADOUT.

but its benefits are unevenly distributed across evaluation conditions.

RQ2: Matryoshka dimensions. Matryoshka Representation Learning (MRL; Kusupati et al., 2022) provides a stricter setting than standard full-dimensional retrieval because the same model must support multiple nested prefix representations. A regularizer that helps only at the final embedding dimension may still be poorly aligned with compressed retrieval views. We therefore use MRL to test whether the spread-out benefit remains visible when retrieval is performed from low-dimensional prefixes, and Table 2 reports retrieval scores by prefix dimension.

Across the MRL results, SPREADOUT improves most prefix dimensions, but the pattern is not strictly monotonic across models. For ModernBERT-base and Llama-3.2-1B, the smallest prefix receives one of the largest relative gains, suggesting that spread-out can help compressed retrieval views. Qwen3-0.6B-Base shows a slightly different pattern: the 64-dimensional prefix changes marginally, while improvements appear consistently from 128 dimensions onward and persist at the full dimension. Taken together, these results suggest that spread-out regularization is compatible with Matryoshka training, while its benefit at each prefix depends on the interaction between prefix capacity and model architecture.

Finally, we check whether the observed gains depend on a narrow optimization setting. We conduct two additional evaluations: varying the learning rate and repeating training with three random seeds. The learning-rate check tests whether the result is tied to a single tuning choice, while the seed check tests whether it is driven by one favorable training run. The full robustness results are reported in Appendix E. These checks indicate that the spread-out loss is relatively robust to both learning-rate variation and random-seed variation.

5 Analysis

In the previous section, we observe that SPREADOUT improves retrieval in our evaluated settings. In this section, we further probe where this improvement occurs by asking: *Which query conditions benefit most from spread-out regularization, and what representation-level changes accompany these gains?* We conduct this analysis across the three evaluated model architectures.

5.1 Query-Length Breakdown

Because the evaluated benchmarks differ in task purpose, query format, passage format, and length distribution, we further analyze performance through the lens of query length. We group evaluation test-set queries into three bins: short (<100 tokens), medium (100–200), and long (>200). The bins are highly imbalanced, with short queries accounting for most evaluation examples. To avoid making the bin-level comparison sensitive to this

Model	Bin	CON	SP	Δ%
ModernBERT-base	Short (<100)	33.69	35.40	+5.06
	Med (100–200)	57.04	54.06	−5.22
	Long (>200)	42.25	41.76	−1.17
	Overall	35.91	37.13	+3.37
Qwen3-0.6B-Base	Short (<100)	51.13	51.81	+1.32
	Med (100–200)	66.28	65.74	−0.82
	Long (>200)	49.52	49.33	−0.39
	Overall	53.47	53.68	+0.40
Llama-3.2-1B	Short (<100)	43.59	44.41	+1.89
	Med (100–200)	69.53	69.44	−0.13
	Long (>200)	54.30	53.01	−2.38
	Overall	51.26	51.34	+0.16

Table 3: Retrieval nDCG@10 by model and query-length bin. CON = CONTRASTIVE and SP = SPREADOUT. Overall is computed on the same capped samples; Appendix F provides details. Δ% is relative to CONTRASTIVE.

imbalance, we use a balanced subsampling protocol: in each run, we cap the number of sampled examples in the short and medium bins by the number of available long-query examples, compute retrieval performance within each bin, and repeat this process 10 times. Table 3 reports the averaged results across these 10 samples, with additional details on the Overall row in Appendix F.

Table 3 shows that the benefit of SPREADOUT is concentrated in the short-query bin across all three architectures. The short-query score improves for ModernBERT-base (+5.06%), Qwen3-0.6B-Base (+1.32%), and Llama-3.2-1B (+1.89%). By contrast, the medium and long bins show smaller or negative changes, with the largest declines appearing for ModernBERT-base on medium queries and for Llama-3.2-1B on long queries. As a result, the overall balanced-bin gains remain positive but more modest for the larger decoder-only models.

This query-length pattern is consistent with the per-task breakdown in Table 1. Several short-query tasks, such as FEVER, HotpotQA, and LegalSummarization, show clear gains under SPREADOUT, while longer-query tasks such as ArguAna show smaller or less consistent changes. These results suggest that spread-out regularization is not uniformly beneficial across query conditions; rather, its retrieval benefit is most reliable when queries are short and under-specified.

5.2 Geometry Diagnostics

To characterize the distribution of embeddings, we use two complementary metrics. **Mean off-diagonal cosine** ($\overline{\cos}_{\text{off}}$) measures the average pairwise directional concentration:

$$\overline{\cos}_{\text{off}} = \frac{1}{N(N-1)} \sum_{i \neq j} s(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

where N is the number of embeddings in the evaluation set.

Squared off-diagonal cosine (s_2) measures the average squared pairwise directional similarity:

$$s_2 = \frac{1}{N(N-1)} \sum_{i \neq j} s(\mathbf{x}_i, \mathbf{x}_j)^2. \tag{4}$$

Lower $\overline{\cos}_{\text{off}}$ indicates weaker average directional alignment, while lower s_2 indicates fewer large pairwise alignments regardless of sign.

Table 4 presents these metrics for query embeddings from three representative architectures, computed on retrieval-task test samples. Following the same balanced subsampling protocol as Section 5.1, we report the raw metric values under CONTRASTIVE and SPREADOUT averaged over 10 random subsamples.

Across all three models, the short-query bin shows the clearest reduction in both $\overline{\cos}_{\text{off}}$ and s_2 under SPREADOUT. The magnitude and direction of the remaining changes are architecture dependent. For ModernBERT-base, SPREADOUT reduces both metrics in all length bins, with the largest reduction for short queries. For Qwen3-0.6B-Base, the short bin decreases substantially, while medium and long bins increase. For Llama-3.2-1B, short queries decrease clearly, medium queries decrease modestly, and long queries change only marginally.

These diagnostics offer one possible account for the concentration of retrieval gains on short queries. Short queries lack sufficient disambiguating context, which may encourage the encoder to map semantically diverse queries into a narrow cone. The spread-out penalty appears to counteract this query-side crowding. Because retrieval gains concentrate on short queries even when geometry changes are observed in other bins, dispersion alone is not a complete explanation; its benefit appears strongest for short, under-specified queries.

6 QUARTO: Query-Length Adaptive Spread-Out

6.1 From Diagnosis to Prescription

The preceding analysis identifies a specific regime in which spread-out is most useful rather than

Model	Condition	$\overline{\cos}$			s_2		
		Short	**Med**	**Long**	**Short**	**Med**	**Long**
ModernBERT-base	CONTRASTIVE	0.372	0.315	0.302	0.151	0.106	0.106
	SPREADOUT	0.187	0.221	0.207	0.046	0.057	0.058
	$\Delta\%$	-49.77	-29.78	-31.42	-69.64	-45.90	-44.99
Qwen3-0.6B-Base	CONTRASTIVE	0.191	0.182	0.163	0.049	0.038	0.040
	SPREADOUT	0.168	0.201	0.179	0.041	0.046	0.045
	$\Delta\%$	-11.85	+10.64	+9.64	-16.93	+19.30	+12.42
Llama-3.2-1B	CONTRASTIVE	0.299	0.161	0.167	0.104	0.032	0.043
	SPREADOUT	0.286	0.157	0.168	0.097	0.031	0.043
	$\Delta\%$	-4.48	-2.23	+0.69	-7.08	-4.63	-1.45

Table 4: Query-side embedding geometry by query length. Raw metric values are averaged over 10 random subsamples using the same balanced subsampling protocol as Table 3. $\Delta\%$ reports the signed relative change from CONTRASTIVE to SPREADOUT, i.e., (SPREADOUT − CONTRASTIVE)/CONTRASTIVE. $\overline{\cos}$ = mean off-diagonal cosine; s_2 = squared off-diagonal cosine. Negative $\Delta\%$ indicates that SPREADOUT lowers the metric.

showing a uniformly beneficial regularization effect. Across architectures, the retrieval gain concentrates in the short-query bin, while medium and long queries show smaller, marginal, or negative changes. The geometry diagnostics provide a compatible explanation: short queries exhibit stronger query-side crowding, and SPREADOUT most consistently reduces this crowding in the short-query regime.

This diagnosis suggests that the relevant question is not whether spread-out pressure should be increased globally, but where that pressure should be concentrated. It leads to a length-adaptive prescription: instead of applying one fixed weight to all queries, QUARTO preserves stronger dispersion for short, under-specified queries and relaxes it where additional dispersion is less consistently helpful.

6.2 Formulation

QUARTO computes a per-query weight from the token length of each query. For each query q_i in a batch with token length l_i, the weight is:

$$w_i = \frac{\log_{10}(r)}{\log_{10}(l_i)} \tag{5}$$

where r is a reference length. Short queries ($l_i < r$) receive weights $w_i > 1$, increasing spread-out pressure; long queries ($l_i > r$) receive $w_i < 1$, reducing it. The batch-level objective becomes:

$$\mathcal{L} = \mathcal{L}_{\text{hard}} + \bar{w} \cdot \mathcal{L}_{\text{so}}, \quad \bar{w} = \frac{1}{B}\sum_{i=1}^{B} w_i \tag{6}$$

The batch mean $\bar{w}$ keeps QUARTO as a global scaling of the same two-term spread-out penalty within each batch, rather than changing the pair structure of $\mathcal{L}_{\text{so}}$. The reference length r is a hyperparameter that controls the crossover point between up-weighting and down-weighting. The implementation details are reported in Appendix G.

The logarithmic form makes this control smooth and sublinear: the crossover $w_i = 1$ occurs at $l_i = r$, while deviations from r change the penalty gradually rather than through a discontinuous threshold or a rapidly growing linear scale. Thus, r should be read as a controllable reference point for the weighting schedule rather than a hard boundary between query types.

Model	CONTRASTIVE	SPREADOUT	QUARTO
ModernBERT-base	33.49	35.37 (+1.88)	38.11 (+4.61)
Qwen3-0.6B-Base	47.73	48.89 (+1.16)	49.19 (+1.47)
Llama-3.2-1B	40.23	40.80 (+0.57)	41.54 (+1.31)

Table 5: 13-task Overall nDCG@10 comparison for CONTRASTIVE, SPREADOUT, and QUARTO across three architectures. Overall denotes the 13-task macro average; parenthesized values report absolute gains over CONTRASTIVE.

6.3 Results

Table 5 presents the 13-task Overall comparison for QUARTO alongside CONTRASTIVE and SPREADOUT. Appendix H reports the full per-task results.

For ModernBERT-base, the largest QUARTO improvements over fixed SPREADOUT occur on short, keyword-style tasks: FEVER improves from 60.46 to 72.26, TRECCOVID from 53.73 to 59.62, and ClimateFEVER from 18.65 to 23.99. QUARTO can also recover cases where fixed SPREADOUT falls below the CONTRASTIVE baseline: on ModernBERT-base, CQADupstackGam-

ing changes from 43.36 to 40.91 to 44.38, and Touché2020 from 34.40 to 31.74 to 35.49 under CONTRASTIVE, SPREADOUT, and QUARTO, respectively.

These examples are consistent with the design of QUARTO: it preserves stronger dispersion pressure for short queries while reducing unnecessary pressure on longer inputs, where fixed spread-out regularization is less reliably helpful.

We also check two configuration choices tied to QUARTO: Table 12 sweeps the reference length over $r \in \{64, 128, 256\}$, and Table 13 evaluates a length-bucketed batching variant. The reference sweep shows that the main pattern is not tied to a single crossover value, while length-bucketed batching changes the overall scores only modestly.

7 Discussion

Taken together, our experiments suggest a conditional view of spread-out regularization for dense retrieval. Across architectures and training settings, SPREADOUT improves overall retrieval, but the gain is concentrated in short-query conditions and becomes smaller, mixed, or negative for medium and long queries. Thus, spread-out should not be understood as a universally beneficial increase in representation dispersion; it is most useful when it corrects a specific query-side crowding pattern, which QUARTO turns into a simple length-adaptive prescription.

Conditional uniformity. This finding refines the usual alignment–uniformity view of contrastive representation learning. Prior work shows that contrastive objectives encourage positive-pair alignment and hyperspherical uniformity, and that representation quality is tied to avoiding collapsed or anisotropic embedding spaces (Wang and Isola, 2020; Gao et al., 2021; Ethayarajh, 2019). Our results are consistent with this view, but they also show that more dispersion is not automatically better for retrieval: applying the same repulsive pressure to all queries can help when representations are crowded, but can become unnecessary or harmful when the query already provides enough semantic detail.

Query length as a diagnostic signal. We do not interpret query length as the causal factor by itself. Instead, length serves as an observable proxy for a broader retrieval condition: short queries are often less specified, more ambiguous, and more dependent on the encoder to separate fine-grained intents from limited lexical evidence. This interpretation is aligned with work on query performance prediction, where query clarity characterizes ambiguity and expected retrieval difficulty (Cronen-Townsend et al., 2002). The short-query pattern should therefore be read as evidence that spread-out is most useful for under-specified queries, not as a claim that all short queries intrinsically benefit.

Geometry as a mechanism, not an objective. The geometry analysis provides a plausible mechanism for the observed retrieval gains, but it should not be treated as the final optimization target. Lower pairwise cosine concentration indicates broader query-side dispersion, yet retrieval quality ultimately depends on whether this change improves ranking. The useful intervention is therefore not simply to maximize dispersion, but to apply dispersion pressure where the representation geometry appears to limit query discrimination.

Practical implications. For retrieval systems dominated by short, keyword-style, or under-specified queries, stronger query-side dispersion can be useful; for verbose, legal, argumentative, or otherwise highly specified queries, the same global pressure should be applied more cautiously because it may disrupt alignment. More broadly, our results suggest that text-embedding regularization should be guided by input-level diagnostics rather than aggregate benchmark averages alone.

8 Conclusion

This paper presents a controlled empirical study of spread-out regularization for dense text embeddings. We find that spread-out can improve retrieval, but its benefit is conditional rather than uniform: gains concentrate on short, under-specified queries where query embeddings exhibit stronger directional crowding. This diagnosis motivates QUARTO, a simple length-adaptive objective that concentrates dispersion pressure where it is most useful instead of applying it uniformly to all queries. Overall, our results suggest that embedding regularization should be guided by input-level diagnostics, linking geometric objectives to the retrieval conditions under which they actually improve ranking.

Ethical Considerations

This work does not collect new user data, recruit human annotators, or deploy a user-facing system. It studies training objectives for retrieval models using existing public research artifacts and aggregate benchmark metrics. Because the source corpora and benchmarks include web, community QA, legal, biomedical, and fact-verification text, they may contain naturally occurring sensitive, personal, or offensive content. Improved retrievers may also be incorporated into downstream search or retrieval-augmented generation systems, whose behavior depends on corpus quality, bias, and coverage. The results should therefore not be read as validating deployment in high-stakes settings without domain-specific evaluation, corpus auditing, and safety analysis. Additional artifact, compute, software, and AI-assistance details are reported in Appendix C.

Limitations

This study focuses on a controlled retrieval setting in which spread-out regularization can be isolated from other training changes. This design makes the comparison interpretable, while leaving two natural extensions for future work.

First, QUARTO uses a simple log-based weighting rule with fixed clipping bounds and reference lengths. This choice is transparent and easy to reproduce, while leaving room for learned, task-conditioned, or distribution-aware weighting schemes in future work.

Second, although our evaluated models cover both encoder-only and decoder-only architectures, the largest backbone is Llama-3.2-1B (1.2B parameters). Future work should test heavier retrieval backbones, such as Qwen3-Embedding-4B and NV-Embed-style LLM retrievers (Zhang et al., 2025; Lee et al., 2025), and examine whether the same query-length patterns hold for reasoning-oriented retrieval.

References

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. 2026. jina-embeddings-v5-text: Task-targeted embedding distillation. *Preprint*, arXiv:2602.15547.

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Overview of the fire 2019 aila track: Artificial intelligence for legal assistance. In *Fire*.

Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Christian Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of touché 2020: Argument retrieval. In *Conference and Labs of the Evaluation Forum*.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 299–306, New York, NY, USA. Association for Computing Machinery.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. Climate-fever: A dataset for verification of real-world climate claims. *Preprint*, arXiv:2012.00614.

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, Hong Kong, China. Association for Computational Linguistics.

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*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium (ADCS)*, ADCS '15, pages 3:1–3:8, New York, NY, USA. ACM.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2022. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.

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. *ArXiv*, abs/2308.03281.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. *Companion Proceedings of the The Web Conference 2018*.

Laura Manor and Junyi Jessy Li. 2019. Plain English summarization of contracts. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 1–11, Minneapolis, Minnesota. Association for Computational Linguistics.

Meta AI. 2024. Llama 3.2 | model cards and prompt formats.

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.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A human-generated MAchine reading COmprehension dataset.

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

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.

Nandan Thakur, Crystina Zhang, Xueguang Ma, and Jimmy Lin. 2025. Hard negatives, hard lessons: Revisiting training data quality for robust information retrieval with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9064–9083, Suzhou, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hayato Tsukagoshi and Ryohei Sasano. 2025. Redundancy, isotropy, and intrinsic dimensionality of prompt-based text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25915–25930, Vienna, Austria. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

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.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. Trec-covid: Constructing a pandemic information retrieval test collection. *Preprint*, arXiv:2005.04474.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

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 N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

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

Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. 2017. Learning spread-out local feature descriptors. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4605–4613.

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

A RLHN-680K Training Data

The RLHN-680K training distribution is highly skewed toward short queries. Table 6 reports query-length statistics for the training queries and their source subsets, using the same length bins as Section 5.1. We compute lengths with whitespace tokenization for this descriptive analysis. Overall, 99.41% of training queries contain fewer than 100 tokens; the medium and long bins are almost entirely contributed by ArguAna.

B Training Configuration and Result Reporting

Table 7 reports the main training and reporting settings used throughout the experiments. All models are trained on RLHN-680K tuples with one positive and seven hard negatives per query. Unless otherwise stated, reported benchmark scores are from the fixed epoch-2 checkpoint selected by the training schedule, with no task-specific tuning on the evaluation benchmarks. Single-run results are reported as single runs; the robustness appendix separately reports three seed runs and their mean trend. We did not tune hyperparameters on the 13 evaluation benchmarks. The main training recipe was fixed before benchmark evaluation; the learning-rate and seed analyses in Appendix E, together with the reference-length ablation in Appendix G, are used as robustness checks rather than task-specific model selection.

C Responsible Research and Reproducibility Details

Scientific artifacts, licenses, and intended use. This paper uses existing scientific artifacts for research training, evaluation, and analysis. We cite the artifact creators in the main text, related work, and Appendix D, and we do not redistribute the original datasets, model checkpoints, or benchmark corpora as part of this submission. For composite benchmark suites, licenses and access terms are inherited from the original task providers; when license information is not clearly specified for a component artifact, we treat this as a limitation and avoid redistribution. Table 8 summarizes artifact use at the level relevant to this submission.

Data provenance and sensitive content. We do not collect new data or add human annotations. The training and evaluation data are existing research artifacts, but their source domains include web search, community QA, scientific and biomedical retrieval, argument retrieval, fact verification, and legal retrieval. These corpora may contain naturally occurring personal names, sensitive legal or biomedical content, offensive language, or biased coverage from their original sources. Because we use existing public research artifacts, we did not conduct a new independent PII or offensive-content audit or de-identification pass. Our experiments use these artifacts only for aggregate retrieval evaluation and do not attempt to identify individuals, infer sensitive attributes, release original text examples, or release model outputs tied to individual people.

Compute budget and software. Main training jobs run on institutional 4-GPU nodes with NVIDIA H100 GPUs. Across the full study, including reported experiments, ablations, robustness checks, and exploratory runs, total compute usage exceeded 10,000 GPU-hours. The main software versions are pinned in the project environment: PyTorch 2.6.0, Transformers 5.0.0, Datasets 4.5.0, DeepSpeed 0.18.5, Accelerate 1.12.0, FlashAttention 2.7.4, Tokenizers 0.22.2, NumPy 2.3.5, SciPy

Source	N	Share (%)	Mean	P25	P50	P75	P90	<100 (%)	100–200 (%)	>200 (%)
All	648,766	100.00	9.2	5	7	9	14	99.41	0.33	0.25
MS MARCO passage	455,748	70.25	5.9	4	6	7	9	100.00	0.00	0.00
HotpotQA	84,395	13.01	17.8	12	15	21	28	99.99	0.01	0.00
NQ	57,539	8.87	9.2	8	9	10	11	100.00	0.00	0.00
FEVER	28,869	4.45	8.1	6	8	10	12	100.00	0.00	0.00
SCIDOCS-RR	12,654	1.95	9.5	7	9	11	14	100.00	0.00	0.00
FiQA	5,496	0.85	10.8	8	10	13	17	100.00	0.00	0.00
ArguAna	4,065	0.63	202.3	139	182	241	313	6.37	52.99	40.64

Table 6: Query length distribution of the RLHN-680K training data. Share is each source subset's proportion of the training set; length-bin columns report within-row percentages.

Category	Setting	Value
Training data	Tuple format	RLHN-680K query tuples with one positive and seven hard negatives.
Backbones	Model scale	ModernBERT-base (149M), Qwen3-0.6B-Base (596M), and Llama-3.2-1B (1.2B).
Optimization	Optimizer and schedule	AdamW with DeepSpeed ZeRO stage 1, WarmupDecayLR, warmup ratio 0.1, and max grad norm 1.0.
Optimization	Learning rate and epochs	5×10^{-6} for main runs and 2 epochs; the robustness LR check uses 1×10^{-5}.
Batching	Batch construction	4 GPUs, per-device batch size 16, gradient accumulation 2, train group size 8.
Inputs	Length limits	Query max length 512; passage max length 512.
Embeddings	Pooling and normalization	Last-token pooling, L2-normalized embeddings, temperature 0.05, and bf16 mixed precision.
Spread-out	Regularization settings	Fixed SPREADOUT uses coefficient 1.0; QUARTO clips weights to [0.2, 4.0].
Checkpoints	Reporting protocol	Checkpoints are saved every 1690 steps and at the end of each epoch; results report nDCG@10 per task and macro average over the 13-task suite.

Table 7: Training, model, and reporting settings used in the main experiments.

1.17.0, Hugging Face Hub 1.3.5, MTEB 2.12.26, Sentence-Transformers 5.4.1, FAISS CPU 1.13.2, and pytrec eval 0.5.

Human subjects and AI assistance. This work does not involve human subjects, crowdworkers, or newly recruited annotators. AI assistants were used to support language polishing, editing, checklist preparation, and coding and debugging. They were not used as authors, did not independently generate experimental results, and did not verify scientific claims. The authors reviewed and remain responsible for all text, analyses, citations, results, and conclusions.

D Evaluation Suite Details

The 13-task retrieval suite is selected to cover both domain diversity and query-length diversity. By domain, it includes fact verification benchmarks (ClimateFEVER and FEVER), community or question-answering retrieval benchmarks (CQADupstackGaming, CQADupstackUnix, FiQA2018, and HotpotQA), scientific and biomedical retrieval benchmarks (SCIDOCS and TRECCOVID), argument retrieval benchmarks (ArguAna and Touché2020), and legal retrieval benchmarks (LegalSummarization, AILACasedocs, and AILAStatutes). All evaluations use the standard MTEB task definitions and train-test splits. We do not construct additional splits or perform extra cross-dataset de-duplication between RLHN-680K and the evaluation benchmarks beyond the filtering and split definitions provided by the source artifacts. We evaluate with the MTEB retrieval pipeline (mteb==2.12.26), using the default task splits, corpora, and qrels unless otherwise stated. Retrieval scores use L2-normalized embeddings and nDCG@10 as the primary metric; Table 1 and related summary tables report macro averages across the 13 tasks without task-specific weighting. Table 9 summarizes the role of each benchmark.

Table 14 reports the test-query counts used in the evaluation. Counts are computed after MTEB qrel filtering, so each listed query has at least one positive document.

By query length, the suite includes ten benchmarks dominated by short queries: ClimateFEVER, CQADupstackGaming, CQADupstackUnix, FEVER, FiQA2018, HotpotQA, LegalSummarization, SCIDOCS, Touché2020, and TRECCOVID and three benchmarks with substantially longer queries: AILACasedocs, AILAStatutes, and ArguAna. Table 10 reports the query length distribution for each benchmark.

E Robustness Check

Table 11 reports additional ModernBERT-base robustness checks on the 13-task retrieval suite. The learning-rate check uses 1×10^{-5}, while the random-seed check repeats the main setting with three independent seeds using the same benchmark suite and metric as Table 1. SPREADOUT improves Overall under the alternative learning rate (28.45 to 32.78) and in all three seed runs; the seed mean increases from 34.70 for CONTRASTIVE to 36.47 for

Artifact	Role	License and terms	Use and redistribution note
RLHN-680K	Training data	Creative Commons Attribution Share Alike 4.0	Used for research fine-tuning of retrieval models, not redistributed
MTEB retrieval tasks and component datasets	Evaluation benchmarks	MTEB and component-task terms	Used for benchmark evaluation under original task providers' terms.
ModernBERT-base	Model backbone	Apache License 2.0	Used as a publicly released checkpoint for research fine-tuning and analysis.
Qwen3-0.6B-Base	Model backbone	Apache License 2.0	Used as a publicly released checkpoint for research fine-tuning and analysis.
Llama-3.2-1B	Model backbone	Llama 3.2 Community License Agreement	Used under the provider's model terms for research fine-tuning and analysis.
Training and evaluation software	Implementation	Open-source package licenses	Used for training, evaluation, and result aggregation.

Table 8: Scientific artifacts used in this work and their intended use in the submission.

Benchmark	Domain	Description
AILACasedocs (Bhattacharya et al., 2019)	Legal retrieval	Legal scenario retrieval against relevant case documents.
AILAStatutes (Bhattacharya et al., 2019)	Legal retrieval	Legal scenario retrieval against relevant statutes.
ArguAna (Wachsmuth et al., 2018)	Argument retrieval	Argument-to-counterargument retrieval with longer argumentative queries.
ClimateFEVER (Diggelmann et al., 2021)	Fact verification	Climate-claim retrieval against evidence passages.
CQADupstackGaming (Hoogeveen et al., 2015)	Community QA	Duplicate-question retrieval from the Gaming Stack Exchange forum.
CQADupstackUnix (Hoogeveen et al., 2015)	Community QA	Duplicate-question retrieval from the Unix Stack Exchange forum.
FEVER (Thorne et al., 2018)	Fact verification	Claim retrieval against Wikipedia evidence.
FiQA2018 (Maia et al., 2018)	Finance QA	Financial question retrieval over answer passages.
HotpotQA (Yang et al., 2018)	Multi-hop QA	Question retrieval requiring evidence from multiple supporting passages.
LegalSummarization (Manor and Li, 2019)	Legal retrieval	Summary or plain-English query retrieval against full legal texts.
SCIDOCS (Cohan et al., 2020)	Scientific retrieval	Scientific document matching and related-paper retrieval.
Touché2020 (Bondarenko et al., 2020)	Argument retrieval	Retrieval for controversial and argumentative topics.
TRECCOVID (Voorhees et al., 2020)	Biomedical retrieval	COVID-19 biomedical literature retrieval.

Table 9: Descriptions of the 13 retrieval benchmarks used in the evaluation suite.

SPREADOUT. These results show that the SPREADOUT gain is not tied to a single training configuration, but remains consistent across learning-rate and random-seed variations.

F Analysis Sampling and Query Visualizations

Query-length sampling and Overall row. For Table 3, each random subsample draws evaluation test queries within the short, medium, and long bins described in Section 5.1. Because long-query bins are the smallest, the short and medium bins are capped by the number of available long-query examples before computing bin-level retrieval scores. The bin rows report the average nDCG@10 within each bin over 10 subsamples. The Overall row is recomputed on the union of the sampled short, medium, and long queries in each subsample and then averaged over the 10 subsamples; it is therefore query-weighted within the capped sample, not the raw full-test-set Overall from Table 1.

Figure 2 compares ModernBERT-base query embedding PCA views under CONTRASTIVE and SPREADOUT for short and non-short query groups. The SPREADOUT points occupy a visibly broader region than the CONTRASTIVE points, matching the query-side dispersion pattern measured in Table 4. Figure 3 adds a complementary single-view 3D PCA comparison between the CONTRASTIVE and SPREADOUT ModernBERT-base query embeddings. Figure 4 shows the same comparison across MRL prefix dimensions. Across dimensions, SPREADOUT generally yields more dispersed query embeddings, with the smallest 64-dimensional prefix showing the clearest separation between CONTRASTIVE and SPREADOUT.

G QUARTO Implementation Details

QUARTO computes a length-dependent spread-out weight from each query length: shorter queries receive stronger spread-out pressure, while longer queries receive weaker pressure. For this training-time weight, query length is measured with the corresponding model tokenizer. To avoid unstable weights for extremely short queries, we use an effective length floor of 10 tokens, i.e., $\tilde{l}_i = \max(l_i, 10)$, before applying Eq. 5. We clip this weight to $[0.2, 4.0]$. For models with final embedding dimension below 1024, we use reference length 128; for models with final embedding dimension at least 1024, we use reference length 256. Thus, ModernBERT-base ($D = 768$) uses reference length 128, while Qwen3-0.6B-Base ($D = 1024$) and Llama-3.2-1B ($D = 2048$) use reference length 256.

Table 12 reports a reference-length ablation for QUARTO over $r \in \{64, 128, 256\}$.

Table 13 reports an exploratory length-bucketed batching check. In this variant, training examples are sorted by query length, grouped into local buckets before mini-batch sampling, and then shuffled at the bucket level. This changes the length composition of updates while leaving the QUARTO objective unchanged. Across models, the Overall changes are modest, suggesting that the main conclusions are not driven by this batching choice.

Benchmark	Mean	P25	P50	P75	P90
AILACasedocs	652.8	428	648	810	995
AILAStatutes	652.8	428	648	810	995
ArguAna	247.4	170	222	296	384
ClimateFEVER	27.0	18	25	33	43
CQADupstackGaming	13.8	11	13	16	19
CQADupstackUnix	13.2	10	12	16	19
FEVER	13.8	11	13	16	19
FiQA2018	15.7	12	15	19	23
HotpotQA	22.7	18	21	26	32
LegalSummarization	22.1	13	18	25	43
SCIDOCS	16.5	12	16	20	23
Touché2020	10.0	8	10	11	13
TRECCOVID	16.4	13	16	18	23

Table 10: Query length distribution by benchmark.

Check	Setting	Condition	Argu	CQA-G	CQA-U	CFEV	FEV	FiQA	HpQA	SCI	TREC	Tché	Legal	AILA-C	AILA-S	Overall
Learning rate	Alt. LR	CONTRASTIVE	58.17	31.54	19.46	15.79	53.59	11.44	48.38	8.55	33.81	9.70	48.01	13.72	17.68	28.45
	Alt. LR	SPREADOUT	57.28	37.09	23.28	15.54	62.48	18.09	54.68	7.08	32.62	23.44	53.60	17.14	23.84	32.78
Random seed	Seed 1	CONTRASTIVE	56.51	44.22	25.91	18.54	54.61	21.03	48.98	9.08	51.41	35.72	49.99	14.52	17.23	34.44
	Seed 1	SPREADOUT	54.24	42.16	26.47	21.11	70.11	20.41	52.22	10.39	48.81	33.92	53.27	14.89	19.49	35.96
	Seed 2	CONTRASTIVE	56.10	43.36	25.99	18.11	48.85	21.32	46.65	9.05	49.42	34.40	49.75	14.36	18.09	33.49
	Seed 2	SPREADOUT	53.31	40.91	25.38	18.65	60.46	20.48	50.89	9.38	53.73	31.74	55.19	20.23	19.54	35.37
	Seed 3	CONTRASTIVE	55.14	44.37	25.34	19.48	64.18	21.00	51.00	8.37	55.96	40.87	52.03	13.62	18.69	36.16
	Seed 3	SPREADOUT	54.08	44.38	28.10	19.83	70.97	23.71	54.69	11.17	55.80	42.21	55.46	17.47	16.97	38.06

Table 11: Robustness checks on the 13-task retrieval benchmark suite. Abbreviations and metrics follow Table 1.

Thus, although QUARTO is applied at the batch level and short-query weights are averaged within a mini-batch, this averaging does not appear to substantially change the aggregate retrieval effect.

H Full QUARTO Results

Table 15 reports the full 13-task comparison among CONTRASTIVE, SPREADOUT, and QUARTO. QUARTO improves the 13-task Overall score for all three model architectures. The largest task-level gains appear on short-query-heavy benchmarks, including FEVER, TRECCOVID, Touché2020, CQADupstackGaming, and FiQA2018, depending on the model. These results provide the per-task detail behind the Overall summary in Table 5.

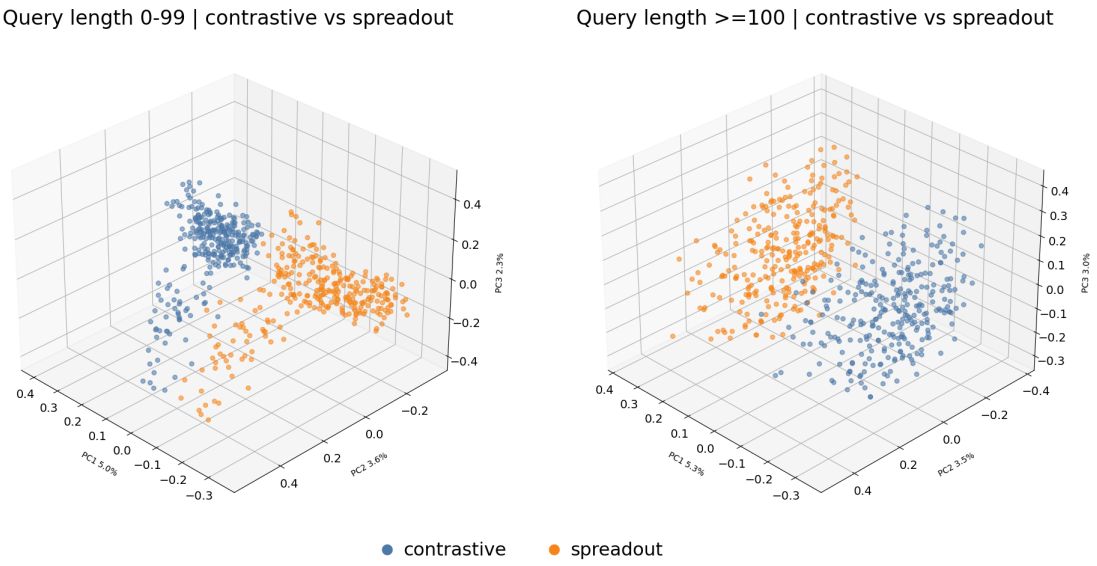

Figure 2: Query embedding PCA views for ModernBERT-base, comparing CONTRASTIVE and SPREADOUT for short (< 100 tokens) and non-short (≥ 100 tokens) query groups.

Model	Ref.	Argu	CQA-G	CQA-U	CFEV	FEV	FiQA	HpQA	SCI	TREC	Tché	Legal	AILA-C	AILA-S	Overall
ModernBERT-base	64	53.14	39.37	24.10	20.18	70.27	20.79	54.86	12.30	50.15	30.99	54.02	15.03	17.72	35.61
	128	54.46	44.38	26.94	23.99	72.26	22.38	53.86	11.13	59.62	35.49	55.76	17.60	17.56	38.11
	256	53.44	41.16	25.92	23.27	72.23	21.19	55.33	12.43	55.06	28.68	55.32	15.69	16.46	36.63
Qwen3-0.6B-Base	64	62.02	53.77	39.75	31.50	83.90	38.79	71.62	21.96	76.90	49.53	61.56	22.13	25.35	49.13
	128	62.81	53.32	39.02	30.37	83.93	38.17	71.14	21.96	76.81	46.85	62.63	22.73	25.68	48.88
	256	62.09	53.94	39.46	31.21	83.78	38.70	71.57	21.84	76.40	50.30	61.57	23.51	25.16	49.19
Llama-3.2-1B	64	60.23	46.57	34.02	22.79	83.65	27.35	63.70	19.79	36.33	25.58	57.60	25.75	28.70	40.93
	128	67.13	44.82	33.19	27.40	79.05	33.02	67.02	19.33	24.18	30.29	58.77	24.06	28.38	41.28
	256	66.98	45.14	33.35	27.65	79.21	33.12	67.24	19.15	23.89	29.55	59.18	25.05	30.54	41.54

Table 12: Reference-length ablation for QUARTO on the 13-task retrieval benchmark suite. Abbreviations and metrics follow Table 1.

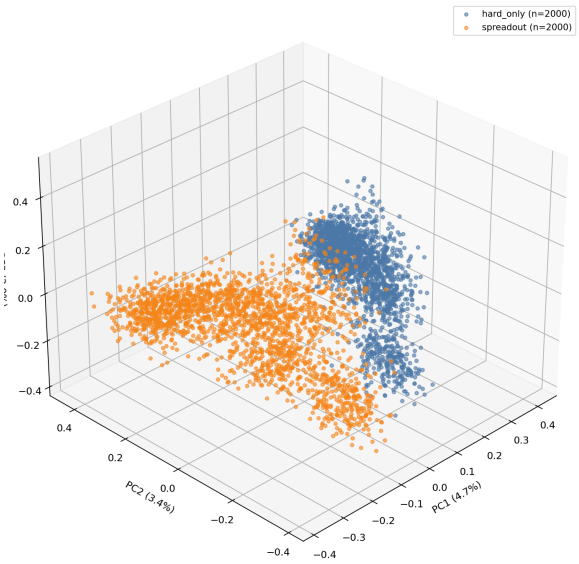

Figure 3: 3D PCA view of ModernBERT-base query embeddings, comparing CONTRASTIVE and SPREADOUT.

Model	Condition	Argu	CQA-G	CQA-U	CFEV	FEV	FiQA	HpQA	SCI	TREC	Tché	Legal	AILA-C	AILA-S	Overall
ModernBERT-base	QUARTO	54.46	44.38	26.94	23.99	72.26	22.38	53.86	11.13	59.62	35.49	55.76	17.60	17.56	38.11
ModernBERT-base	QUARTO + bucket	52.51	42.64	26.99	22.14	73.55	22.73	54.31	11.69	55.44	46.43	54.14	17.96	18.52	38.39
Qwen3-0.6B-Base	QUARTO	62.09	53.94	39.46	31.21	83.78	38.70	71.57	21.84	76.40	50.30	61.57	23.51	25.16	49.19
Qwen3-0.6B-Base	QUARTO + bucket	63.74	52.41	38.98	29.27	85.86	35.46	69.67	22.24	74.68	45.55	62.50	22.31	27.33	48.46
Llama-3.2-1B	QUARTO	66.98	45.14	33.35	27.65	79.21	33.12	67.24	19.15	23.89	29.55	59.18	25.05	30.54	41.54
Llama-3.2-1B	QUARTO + bucket	67.14	47.00	34.65	25.24	82.53	30.48	68.01	20.76	31.15	18.10	57.83	23.73	28.38	41.15

Table 13: Exploratory length-bucketed training results on the 13-task retrieval benchmark suite. Bucketed rows use the same reference length as the corresponding main QUARTO run and the same reporting format as Table 1.

Benchmark	Split	#Queries
AILACasedocs	test	50
AILAStatutes	test	50
ArguAna	test	1,401
ClimateFEVER	test	1,000
CQADupstackGaming	test	1,595
CQADupstackUnix	test	1,072
FEVER	test	1,000
FiQA2018	test	648
HotpotQA	test	1,000
LegalSummarization	test	284
SCIDOCS	test	1,000
Touché2020	test	49
TRECCOVID	test	50

Table 14: Test-query counts for the 13 retrieval benchmarks. Counts include test queries with at least one positive passage after MTEB qrel filtering.

Model	Condition	Argu	CQA-G	CQA-U	CFEV	FEV	FiQA	HpQA	SCI	TREC	Tché	Legal	AILA-C	AILA-S	Overall
ModernBERT-base	CONTRASTIVE	56.10	43.36	25.99	18.11	48.85	21.32	46.65	9.05	49.42	34.40	49.75	14.36	18.09	33.49
	SPREADOUT	53.31	40.91	25.38	18.65	60.46	20.48	50.89	9.38	53.73	31.74	55.19	20.23	19.54	35.37
	QUARTO	54.46	44.38	26.94	23.99	72.26	22.38	53.86	11.13	59.62	35.49	55.76	17.60	17.56	38.11
Qwen3-0.6B-Base	CONTRASTIVE	64.07	51.26	37.94	29.44	83.34	38.35	69.65	21.78	70.49	45.82	62.40	20.77	25.15	47.73
	SPREADOUT	63.30	52.88	38.85	30.01	84.56	38.14	71.42	21.60	77.45	47.47	62.81	21.67	25.45	48.89
	QUARTO	62.09	53.94	39.46	31.21	83.78	38.70	71.57	21.84	76.40	50.30	61.57	23.51	25.16	49.19
Llama-3.2-1B	CONTRASTIVE	68.34	40.64	34.05	29.74	75.57	27.90	65.96	18.20	27.30	20.41	57.87	28.02	29.03	40.23
	SPREADOUT	67.40	42.22	33.42	27.35	77.23	30.20	67.19	19.00	26.12	26.78	59.60	26.81	27.06	40.80
	QUARTO	66.98	45.14	33.35	27.65	79.21	33.12	67.24	19.15	23.89	29.55	59.18	25.05	30.54	41.54

Table 15: Full 13-task nDCG@10 comparison for CONTRASTIVE, SPREADOUT, and QUARTO across three architectures. Abbreviations as in Table 1. Overall denotes the 13-task macro average.

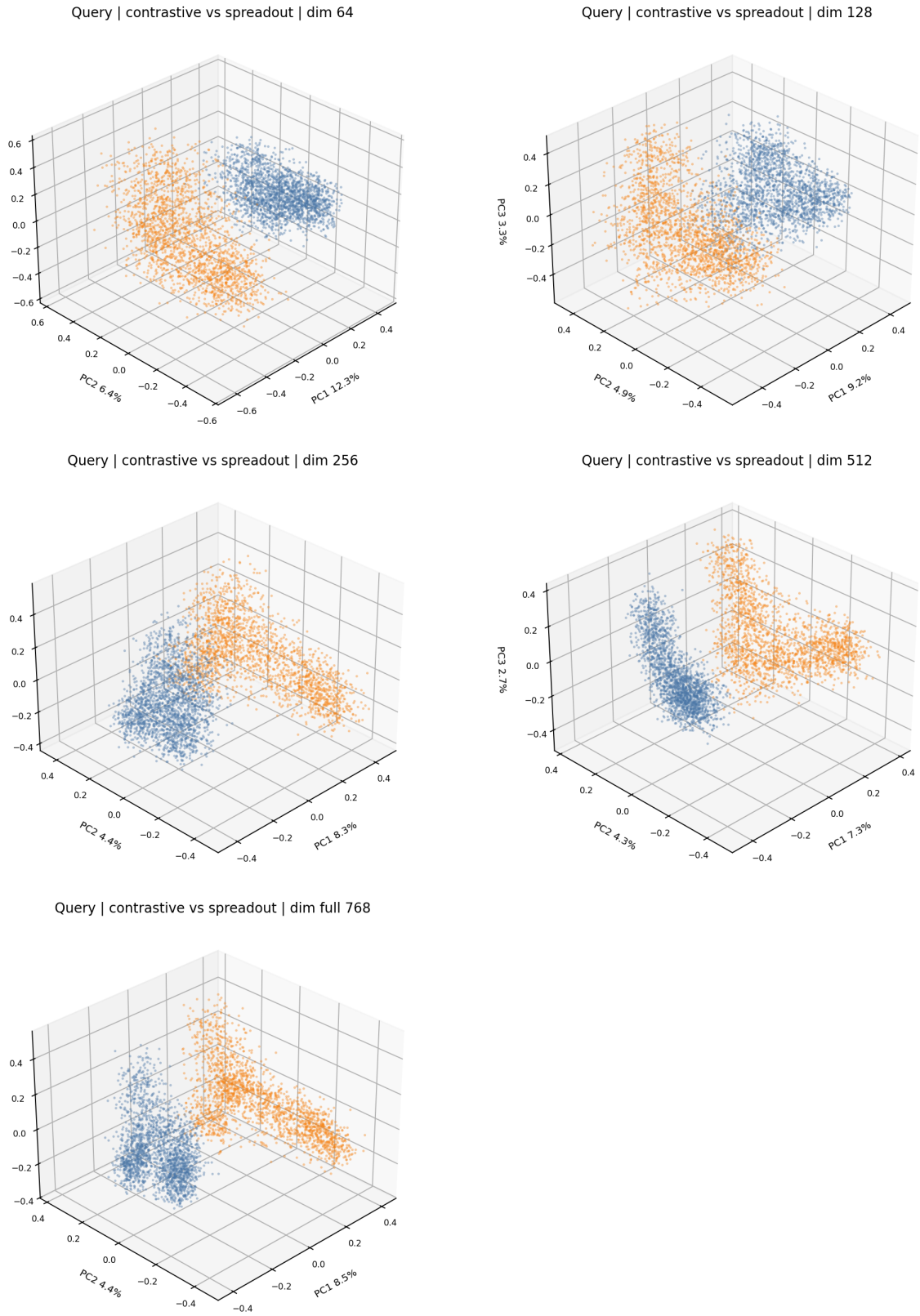

Figure 4: Query embedding PCA views for ModernBERT-base under MRL training, comparing CONTRASTIVE and SPREADOUT across prefix dimensions.