Distilling Large Embeddings via Hyperspherical Householder Quantization

Anonymous ACL submission

Abstract

Large embedding models have become the backbone of modern retrieval systems, offering strong semantic representations at the cost of substantial storage and computation. While recent work explores quantizing embeddings into discrete document identifiers for generative retrieval, most existing approaches rely on Euclidean quantization, which is poorly aligned with the angular geometry induced by contrastive embedding training and often requires long identifier sequences to preserve semantic fidelity. In this work, we propose *Hyperspherical Householder Quantization* (HHQ), a geometry-aware distillation method that compresses large embeddings into short discrete representations via iterative Householder transformations on the unit hypersphere. By explicitly preserving cosine similarity at each step, HHQ distills semantic structure into compact identifiers that remain faithful to the original embedding space. To support reliable generation of these identifiers, we introduce constrained supervised fine-tuning and tree-aware dynamic masking to enforce structural validity during training and inference. Experiments on NQ and MS MARCO show that HHQ achieves competitive or superior retrieval performance using only five tokens per document, substantially reducing decoding cost while retaining strong semantic retrieval accuracy.

1 Introduction

Generative information retrieval (GenIR), as a new paradigm in the field of information retrieval, abandons the traditional two-step "index-retrieve" framework. Instead, it directly generates target document identifiers or the content in an end-to-end manner(Tay et al., 2022; Wang et al., 2022). This fundamentally addresses the search-result mismatch issues inherent in dense retrieval, as well as the storage and memory overhead associated with maintaining large-scale vector indexes.

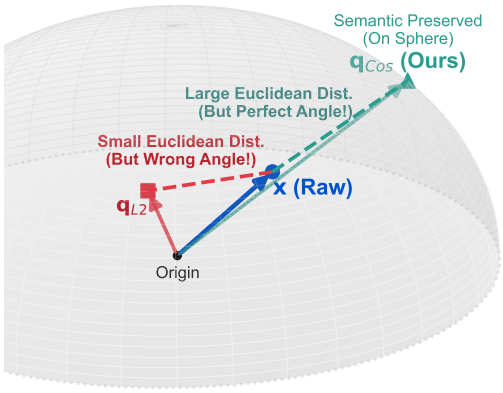

Figure 1: **Euclidean vs. Angular Quantization.** Traditional quantization (red, $\mathbf{q}_{L2}$) minimizes Euclidean distance to the raw embedding $\mathbf{x}$ (blue) but sacrifices the crucial semantic direction. Our method (green, $\mathbf{q}_{Cos}$) preserves angular alignment on the hypersphere, maintaining semantic fidelity essential for contrastive learning.

In GenIR, a transformer-based encoder–decoder is trained to output a document identifier (*docid*) given a query, effectively embedding all document knowledge in the model's parameters. A core challenge is how to design identifiers that capture semantic content and can be predicted accurately from queries (Bevilacqua et al., 2022b). While early attempts used surface-level identifiers (titles, URLs, arbitrary IDs), these are semantically limited and do not scale well (Li et al., 2023).

To address this, researchers have increasingly turned to derive semantically meaningful identifiers from the document embedding space. Consequently, *Vector Quantization* has emerged as a widely adopted technique, primarily due to its capability of imposing a structured and learnable discrete organization upon continuous semantic embeddings. Methods such as Product quantization (PQ) (Zhou et al., 2024b), Residual quantization (Deng et al., 2025), and their variants generate compact codewords that act as discrete document

identifiers(Rajput et al., 2023). However, existing quantization-based identifiers have two fundamental structural limitations.

First, quantization methods often fail to fully utilize the embedding space, requiring either extremely large code-books or long token sequences. As an example, a scheme with $k = 256$ clusters and $m = 24$ PQ code-words yields 256^{24} possible identifiers — which is massively over-sized for a corpus of only $\sim 300\,\text{k}$ documents. As a result, generative models frequently produce invalid IDs, causing inefficiency at both training and inference time. Long identifiers can reduce the code-book size but slow generation (Sun et al., 2023).

Second, there is a **mismatch in similarity metrics** between embedding models and quantization methods. Mainstream embedding models are trained via contrastive learning to optimize **cosine similarity** (Reimers and Gurevych, 2019; Gao et al., 2021), operating on a hyperspherical manifold defined by angular distance. In contrast, quantization methods—including k-means, PQ, and residual quantization—are grounded in **Euclidean (L2) distance**, which assumes a flat geometric space. (Dai et al., 2020; Zhe et al., 2019). As illustrated in Figure 1, this geometric mismatch leads to quantization boundaries that are misaligned with the embedding spaces, ultimately distorting the semantic information encoded by the model.

In this work, we propose a quantization method that aligns the entire quantization process with the geometry of the embedding space. Our **Hyperspherical Householder Quantizer (HHQ)** operates directly on the **unit hypersphere**: at each step, the current point selects a direction from a learned codebook and applies a Householder transformation that reflects the vector across a hyperplane, producing a new point on the sphere. Because Householder transformations are orthogonal and norm-preserving, HHQ maintains unit-length consistency throughout and ensures that each token encodes a meaningful directional adjustment that increases **cosine similarity** with the target embedding.

By explicitly optimizing angular alignment rather than Euclidean reconstruction, HHQ achieves high-fidelity quantization with far fewer tokens than PQ or RQ, yielding compact and semantically coherent identifiers. Beyond the quantizer itself, we introduce two practical components that further enhance generative retrieval: (1) a **constrained supervised fine-tuning (cSFT)** objective that guides the model to follow valid decoding paths in the identifier tree, and (2) a **tree-aware dynamic masking** mechanism that prunes incompatible branches during training, substantially accelerating convergence. We describe these components in the following sections and evaluate their effectiveness on standard generative retrieval benchmarks.

2 Related Work

Dense Retrieval and Embedding Models. Dense retrieval maps queries and documents into a shared vector space and ranks candidates by their embedding similarity. Modern embedding models for retrieval are typically trained with **contrastive learning**: positive query–document pairs are pulled together while negatives are pushed apart, using in-batch negatives or mined hard negatives to shape a discriminative geometry (Reimers and Gurevych, 2019; Gao et al., 2021). Recent large-scale models such as Gemini-Embedding (Lee et al., 2025) and Qwen3-Embedding (Zhang et al., 2025) further adopt **Matryoshka Representation Learning (MRL)**, which trains embeddings to retain semantic quality even when truncated to lower dimensions (Kusupati et al., 2022). This enables multi-scale embeddings that trade off accuracy and storage without retraining. Despite these advances, dense retrieval still requires storing high-dimensional vectors for all documents, motivating research on embedding compression and index-free generative retrieval.

Generative Retrieval and Quantized Semantic IDs. Generative retrieval (GenIR) reframes retrieval as sequence generation: a model directly outputs a document identifier given a query, eliminating the need to store a dense vector index. Early systems such as DSI use arbitrary textual or numeric identifiers, but these IDs lack semantic structure and are difficult for the model to predict reliably (Tay et al., 2022). Subsequent work therefore derives **semantic docids** by quantizing document embeddings. Typical approaches include product quantization (PQ) (Zhou et al., 2024b), residual quantization (RQ) (Deng et al., 2025), and VQ-based autoencoders, which map each embedding to a sequence of discrete codewords. These learned IDs provide semantic regularity, but existing methods usually rely on **Euclidean (L2) partitioning** and often require very large codebooks or long token sequences to achieve sufficient corpus

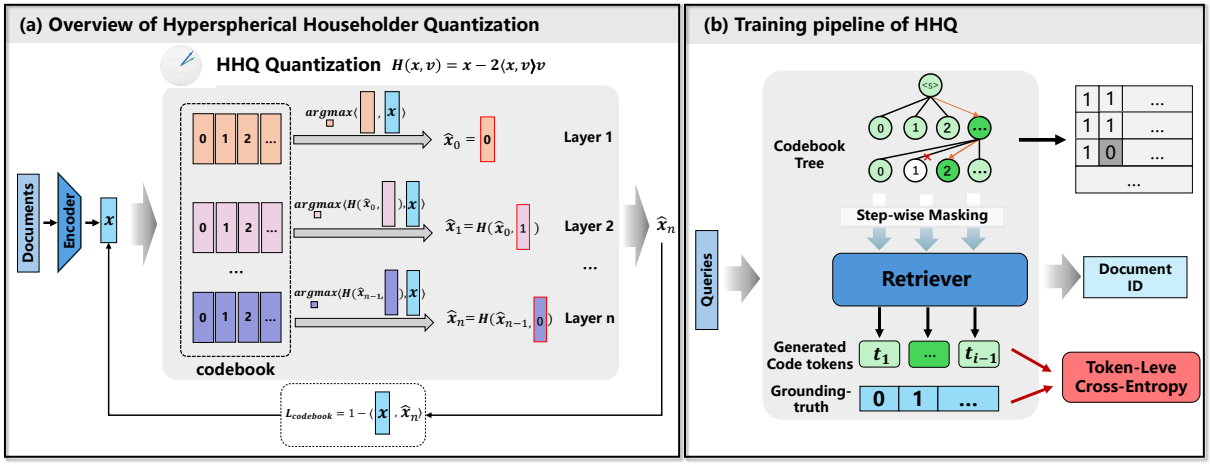

Figure 2: **Overview of Hyperspherical Householder Quantization (HHQ) and its training pipeline.** (a) HHQ iteratively applies Householder reflections guided by a layered codebook to convert document embeddings into compact semantic identifiers while preserving cosine similarity. (b) The resulting identifiers are used to supervise a generative retriever via tree-constrained, token-level cross-entropy with step-wise masking during training.

coverage. This misalignment with contrastively trained embeddings—where semantics are primarily encoded in vector *direction* rather than magnitude—limits the efficiency of current generative retrieval systems. These challenges motivate quantization methods that better match the geometry of modern embedding spaces.

3 Methodology

In this section, we present HHQ, a geometry-aligned framework for distilling continuous embeddings into compact discrete identifiers, and describe how these identifiers are used to train a generative retrieval model. Specifically, we first introduce the training procedure of the hyperspherical quantizer, including codebook initialization and end-to-end optimization. And then describe the inference-time quantization process, followed by the training of a generative model that learns to produce the resulting semantic identifiers.

3.1 Hyperspherical Quantization Mechanism

Given a normalized embedding $\mathbf{x}$, HHQ performs L-layer iterative updates on the unit hypersphere via norm-preserving Householder reflections. Specifically, Each layer i maintains a codebook $V_i = \{\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,K}\}$ of normalized direction vectors. At each step, the quantizer selects a direction from V_i and applies a Householder reflection to progressively align the current approximation $\hat{\mathbf{x}}_{i-1}$ with the target embedding. After L layers, the sequence of selected directions forms the discrete semantic identifier.

$$\mathbf{x} \leftarrow \frac{\mathbf{x}}{\|\mathbf{x}\|_2}, \quad \mathbf{v}_{i,k} \in \mathcal{S}^{D-1}. \tag{1}$$

Initial reflection Direction In the first layer, the quantizer determines the initial reflection direction by identifying the codeword in codebook V_1 that maximizes the cosine similarity with the input vector $\mathbf{x}$:

$$\text{idx}_1 = \arg\max_k \langle \mathbf{x}, \mathbf{v}_{1,k} \rangle. \tag{2}$$

The corresponding codeword, denoted as $\hat{\mathbf{x}}_1 = \mathbf{v}_{1,\text{idx}_1}$, acts as both the first token and the starting point for subsequent Householder refinements.

Iterative Householder Updates. For layers $i > 1$, HHQ refines the approximation by selecting the direction that yields the greatest angular improvement toward the target embedding $\mathbf{x}$. Specifically, given a unit reflection vector $\mathbf{v}$, the Householder reflection acting on a vector $\mathbf{z}$ is defined as:

$$H(\mathbf{z}; \mathbf{v}) = \mathbf{z} - 2\langle \mathbf{z}, \mathbf{v} \rangle \mathbf{v}. \tag{3}$$

Notably, the transformation is norm-preserving ($\|H(\mathbf{z}; \mathbf{v})\| = \|\mathbf{z}\|$) and involutive ($H(H(\mathbf{z}; \mathbf{v}); \mathbf{v}) = \mathbf{z}$).

Then considering the approximation $\hat{\mathbf{x}}_{i-1}$ at layer $(i-1)$ and candidate directions $\{\mathbf{v}_{i,k}\} \subset V_i$ at layer i. The optimal direction index for this i-th layer is determined by maximizing the inner product between the reflection vector and the target $\mathbf{x}$, formulated as:

$$\text{idx}_i = \arg\max_k \langle\, H(\hat{\mathbf{x}}_{i-1}; \mathbf{v}_{i,k}) \,,\, \mathbf{x} \,\rangle \tag{4}$$

Once selected, the actual update is applied:

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i-1} - 2\langle \hat{\mathbf{x}}_{i-1}, \mathbf{v}_{i,\mathrm{idx}_i} \rangle \, \mathbf{v}_{i,\mathrm{idx}_i}, \quad (5)$$

Finally, the updated $\hat{\mathbf{x}}_i$ serves as the new approximation for the next refinement layer.

3.2 Optimization of the Hyperspherical Quantizer

Codebook Initialization. We initialize the codebooks through an offline hierarchical K-means procedure, following a global-to-local refinement scheme:

- Globally, at the first layer, we obtain V_1 by clustering normalized document embeddings, which yields a coarse angular partitioning of the embedding hypersphere.
- Locally, for each deeper layer, we compute the normalized residual $\mathbf{r}_{i-1}$ between the target embedding $\mathbf{x}$ and the current approximation $\hat{\mathbf{x}}_{i-1}$:

$$\mathbf{r}_{i-1} = \frac{\mathbf{x} - \hat{\mathbf{x}}_{i-1}}{\|\mathbf{x} - \hat{\mathbf{x}}_{i-1}\|_2}. \quad (6)$$

  and perform K-means clustering on these residuals to provide layer-specific refinement directions for V_i

Importantly, this initialization serves only as a *directional prior*: it supplies a diverse set of orientations on the sphere but does not constrain learning, as all codebook vectors are updated end-to-end. This allows the subsequent Householder updates to adapt fully to task-specific geometry.

Training Objective. All codebook vectors are trained jointly using the cosine-similarity objective:

$$\mathcal{L} = 1 - \frac{1}{B} \sum_{j=1}^{B} \langle \hat{\mathbf{x}}_L^{(j)}, \mathbf{x}^{(j)} \rangle, \quad (7)$$

where B denotes the batch size, and $\langle \cdot, \cdot \rangle$ is the cosine similarity between $\hat{\mathbf{x}}_L^{(j)}$ and $\mathbf{x}^{(j)}$ (with both vectors ℓ_2-normalized). This loss encourages the sequence of Householder reflections to minimize angular deviation between the reconstructed and target embeddings. This objective directly teaches the quantizer to choose reflection directions that produce the most efficient angular trajectory toward $\mathbf{x}$.

3.3 Constrained Supervised Fine-Tuning for Retrievers

After training the quantizer, we proceed to train a generative model that maps queries to document identifiers. Although each document corresponds to a unique quantized code sequence, the overall code search space is still much larger than the document set, meaning that many possible code paths do not correspond to any real document. To ensure that the model learns to generate only valid identifiers, we introduce a **constrained supervised fine-tuning (cSFT)** strategy based on the hierarchical structure induced by the quantizer. Each prefix restricts the next allowable token to valid descendants in the code tree. Thus, during fine-tuning, the model is explicitly taught to respect the code tree structure by permitting only valid transitions

Codebook Tree Generation. Each semantic identifier produced by HHQ corresponds to a sequence of code tokens $[t_1, t_2, \ldots, t_L]$, which we constitutes a distinct path within the hierarchical codebook tree. To construct this tree, we extract all valid code sequences from a given dataset and compute the conditional probability of each token given its immediate predecessor. These statistical dependencies are then used to build the structured codebook tree.

Constrained Token-Level Cross-Entropy. Given a partial prefix $[t_1, \ldots, t_{i-1}]$, only the children of node t_{i-1} in codebook tree constitute valid choices for the next token. Therefore, we construct a step-wise mask in Finetuning stage.

$$\mathcal{M}_i = \{k \mid k \in ValidChildren(t_{i-1})\}, \quad (8)$$

For a batch of target sequences, we modify the next-token distribution by forcing logits outside the valid set $\mathcal{M}_i$ to $-\infty$, effectively removing them from the softmax support:

$$\tilde{\ell}_{i,k} = \begin{cases} \ell_{i,k}, & \text{if } k \in \mathcal{M}_i, \\ -\infty, & \text{otherwise.} \end{cases} \quad (9)$$

Here, $\ell_{i,k}$ denotes the original logit for token k at position i. The standard cross-entropy loss is then applied over the masked logits:

$$\mathcal{L}_{\text{cSFT}} = -\sum_i \log \frac{\exp(\tilde{\ell}_{i,t_i})}{\sum_{k \in \mathcal{M}_i} \exp(\tilde{\ell}_{i,k})}. \quad (10)$$

Through the above constrained fine-tuning, the model is no longer required to expend learning

capacity on suppressing invalid code paths or memorizing arbitrary exclusion rules. Its generative behavior can more rapidly align with the geometric structure of the target quantizer, thus yielding clearer and more reliable document identifier generation results.

3.4 Quantization and ID Collisions

Our quantizer operates in a compact regime where semantic IDs are not strictly injective: a small fraction of documents may share the same ID. This is an intentional trade-off for efficiency enabled by a compact yet expressive hyperspherical code space. Empirically, we achieve a coverage rate of 0.98, meaning that collisions affect only about 2% of documents. Moreover, these collisions predominantly arise among highly similar embeddings, so the resulting ambiguity is typically semantically coherent and has negligible impact on retrieval quality.

At inference time, the model generates an L-token ID path that resolves to a leaf in the code tree, which may contain one or more documents. When a leaf corresponds to multiple documents, we expand it by sampling (or enumerating, if small) the associated documents as candidates. Since collisions are rare and semantically aligned, this lightweight expansion is sufficient in practice.

4 Experiments

4.1 Datasets and Metrics

Datasets. We evaluate our method on two widely used document retrieval benchmarks: **MS MARCO Document Ranking** and **Natural Questions (NQ)**. These datasets cover both web-scale retrieval and knowledge-intensive question answering. Detailed dataset statistics and construction details are provided in Appendix A.

MS MARCO (Nguyen et al., 2016) Following prior work on generative retrieval (e.g., WebUltron), we construct two 300K-document subsets to evaluate different corpus characteristics. The *Relevant 300K* subset contains documents that have at least one associated relevant query, while the *Random 300K* subset is formed by randomly sampling documents from the full corpus. For both settings, we retain only queries whose relevant documents appear in the corresponding subset.

Natural Questions (NQ) (Kwiatkowski et al., 2019) We adopt the NQ320K setup used in DDRO, where each query is associated with a Wikipedia page. Documents are deduplicated by URL, and the retrieval task requires generating the identifier of the correct page given a query. We follow the predefined train and development splits used in prior work.

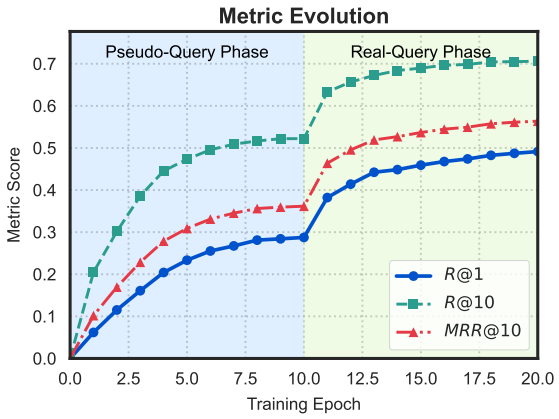

Figure 3: Training dynamics on NQ320K, showing the evolution of R@1, R@10, and MRR@10 across the two-stage training process, with clear gains when transitioning from pseudo-query training to real-query fine-tuning.

Metrics. We follow prior work and report **Recall@1** (R@1), **Recall@10** (R@10), and **MRR@10** to evaluate both the effectiveness of the learned identifier space and the model's ability to generate accurate semantic IDs.

4.2 Baselines.

We compare our method against a broad range of retrieval paradigms to provide a comprehensive evaluation.

Term-based retrieval. We include BM25 (Robertson et al., 1995) and DocT5Query (Nogueira et al., 2019), which represent classical lexical matching and query-expansion–based retrieval. These methods establish the traditional non-neural baseline.

Dense retrieval. DPR (Karpukhin et al., 2020), RepBERT (Zhan et al., 2020), and Sentence-T5 (Ni et al., 2022) are representative dual-encoder models trained with contrastive objectives. They capture semantic similarity through dense embeddings and serve as strong neural baselines.

Generative retrieval (ID-free). SEAL (Bevilacqua et al., 2022a), DynamicRetriever (Zhou et al., 2022), WebUltron (TU) (Zhou et al., 2024b), and ROGER (TU) (Zhou et al., 2024a) generate textual

Model	R@1	R@10	MRR@10
Term-based retrieval			
BM25	14.06	47.93	23.60
DocT5Query	19.07	55.83	29.55
Dense retrieval			
DPR	22.78	68.58	35.92
RepBERT	22.57	65.65	35.13
Sentence-T5	22.51	65.12	34.95
Generative retrieval (ID-free)			
SEAL	29.30	68.53	40.34
DynR.	22.63	68.76	36.08
Ultron (TU)	33.78	67.05	42.51
ROGER (TU)	35.90	69.86	44.92
Generative retrieval (ID-based)			
DSI	27.42	56.58	34.31
DSI-QG	30.17	66.37	38.85
NCI	32.69	69.20	42.84
Ultron (SI)	25.64	65.75	37.12
ROGER (SI)	33.20	69.80	43.45
MINDER	31.00	65.79	43.50
LTRGR	32.80	68.74	44.80
DDRO	48.92	67.31	55.51
Ours			
HHQ	48.33	**70.43**	**55.79**

Table 1: Retrieval performance across term-based, dense, and generative retrieval baselines, where Ultron and ROGER include both TU (title–URL) and SI (semantic ID) variants, compared against our RHQ method.

identifiers such as titles, spans, or salient phrases. These approaches do not rely on numerical document IDs and highlight the effectiveness of free-form generation for retrieval.

Generative retrieval (ID-based). We further compare against models that generate structured document identifiers, including DSI (Tay et al., 2022), DSI-QG (Zhuang et al., 2022), NCI (Wang et al., 2022), WebUltron (SI) (Zhou et al., 2024b), ROGER (SI) (Zhou et al., 2024a), MINDER (Li et al., 2023), LTRGR (Li et al., 2024), and DDRO (Mekonnen et al., 2025). These methods map each document to a fixed identifier space and train the model to generate the corresponding ID, providing a direct comparison to our quantization-based identifier design.

4.3 Implementation Details

Pseudo Queries. In addition to the human-written queries from each dataset, we follow DDRO and augment the training data with **pseudo queries** generated by the DocT5Query model [1]. Each document is paired with synthetic queries that describe its content, providing richer supervision for learning semantic identifiers. We adopt the exact pseudo-query generation procedure used in DDRO to ensure full comparability across methods.

Training Procedure. We use **T5-base (Raffel et al., 2020)** as the generative model for producing document identifiers. The model is trained with a two-stage contrastive sequence-to-sequence fine-tuning (cSFT) pipeline. Each stage is optimized for 10 epochs with a learning rate of 1×10^{-3} and a linear learning-rate scheduler. The first stage focuses on learning coarse semantic alignment in the identifier space, while the second stage refines generation fidelity and improves token-level consistency.

Document Embeddings. For efficiency during initialization and quantization, we use **512-dimensional** document embeddings throughout our experiments. This choice offers a favorable trade-off between semantic fidelity and computational cost, and we observe no performance degradation compared to higher-dimensional representations. We adopt **Qwen3-Embedding-8B** as the base embedding model, and further apply Matryoshka Representation Learning (MRL) to obtain truncated representations for efficient quantization. Additional analysis of embedding choice and dimensionality is provided in Appendix B.

Codebook Configuration. Our quantization module uses a **5-layer codebook**, resulting in 5 generated tokens per document identifier. For each layer, we follow the settings of prior work when applicable. On **NQ**, we adopt the same cluster count as DDRO and WebUltron, using **256** centers per level. On **MS MARCO**, due to its larger corpus and higher semantic diversity, we use **512** centers per level. Additional analysis of code utilization under different codebook configurations is provided in Appendix C.

4.4 Experimental Results

Training Dynamics. The two-stage training process yields clear and consistent improvements, as shown in Figure 3. During the pseudo-query phase, the model rapidly acquires coarse semantic alignment, while the transition to real queries produces a sharp gain in R@1, R@10, and MRR@10. This

[1] https://huggingface.co/datasets/kiyam/ddro-pseudo-queries

Model	MS MARCO					
	Relevant 300K			Random 300K		
	R@1	R@10	MRR@10	R@1	R@10	MRR@10
Term-based retrieval						
BM25	18.94	55.07	29.24	43.85	73.81	54.21
DocT5Query	23.27	61.38	34.25	48.21	77.38	57.95
Dense retrieval						
DPR	28.08	73.10	41.40	42.86	75.52	54.16
RepBERT	25.25	69.18	38.48	40.87	72.81	51.09
Sentence-T5	27.23	72.40	40.70	42.26	75.00	53.59
Generative retrieval (ID-free)						
DynR.	29.04	78.59	42.53	44.13	72.93	55.18
Ultron (TU)	28.96	63.86	40.44	38.49	62.90	46.79
Generative retrieval (ID-based)						
DSI	25.74	53.84	33.92	25.01	48.81	32.21
DSI-QG	27.82	60.26	37.45	34.27	56.79	40.93
NCI	28.35	63.85	38.93	36.99	60.16	47.23
Ultron (SI) (24-token ID)	30.32	72.15	44.16	41.27	68.45	52.00
DDRO (24-token ID)	32.92	73.02	45.76	42.06	69.44	52.41
Ours						
HHQ (5-token ID)	26.36	61.51	36.81	40.67	65.48	49.47

Table 2: Retrieval performance on MS MARCO Relevant 300K and Random 300K across term-based, dense, and generative retrieval baselines, where HHQ (5-token ID) is compared against both ID-free and ID-based generative methods including 24-token PQ-based identifiers.

confirms that pseudo queries provide an effective initialization, whereas real queries refine the semantic identifier space for accurate document generation.

Performance and Efficiency. HHQ achieves strong and consistent performance on both NQ320K and MS MARCO Random 300K, demonstrating the effectiveness of hyperspherical Householder quantization in settings where embedding geometry is informative. On NQ320K, HHQ matches state-of-the-art generative retrieval models while using only 5 tokens per identifier, with slightly lower R@1 (-0.59) but higher R@10 ($+3.12$) and MRR@10 ($+0.28$) compared to DDRO.

On MS MARCO Random 300K, HHQ remains competitive while reducing identifier length from 24 tokens to 5 (a $4.8\times$ reduction). Although its R@1 and MRR@10 are modestly lower than DDRO (-1.39 and -2.94, respectively), the substantially shorter identifiers lead to significantly lower decoding and inference cost. These results indicate that HHQ can effectively distill high-quality embeddings into compact identifiers without sacrificing retrieval accuracy.

HHQ performs noticeably worse on the MS MARCO Relevant 300K subset than on NQ and Random 300K. Compared with DDRO, HHQ shows consistent drops across all metrics, indicating that this split poses unique challenges for embedding-based generative retrieval. Importantly, this behavior is not specific to HHQ. Our Qwen3-Embedding analysis reveals that even strong contrastive embedding models exhibit no clear advantage on Relevant 300K, suggesting that the embedding space itself is less discriminative in this setting. Prior work has shown that MS MARCO Relevant 300K contains limited document diversity and tightly coupled query–document pairs, where dense embeddings tend to collapse and struggle to separate highly similar documents effectively (Reimers and Gurevych, 2021). Since HHQ directly relies on the geometry of the underlying embedding space and we do not tune hyperparameters specifically for this split, its performance naturally

Setting	wo/ Pseudo-Query Phase			w/ Pseudo-Query Phase			ΔR@1
	R@1	R@10	MRR@10	R@1	R@10	MRR@10	
Depth 4, Dim 256	39.87	57.99	45.64	44.28	67.57	51.89	+4.41
Depth 5, Dim 256	37.76	57.80	44.09	42.73	67.65	50.78	+4.97
Depth 6, Dim 256	36.13	57.47	42.81	41.95	67.20	49.92	+5.82
Depth 4, Dim 512	40.87	59.81	47.18	46.68	69.83	54.41	+5.81
Depth 5, Dim 512	39.46	60.09	46.23	45.43	70.03	53.65	+5.97
Depth 6, Dim 512	39.76	60.01	46.27	45.25	69.50	53.18	+5.49

Table 3: Ablation study: improvement from pseudo+query two-stage training on NQ.

reflects these characteristics. We therefore view Relevant 300K as a stress-test case for geometry-driven quantization, rather than a representative scenario for its typical operating regime.

Overall, HHQ delivers strong retrieval accuracy on datasets where the embedding geometry remains informative (e.g., NQ and Random 300K), while offering a markedly smaller identifier space and reduced inference overhead. In addition, HHQ attains competitive performance with a two-stage supervised fine-tuning setup, without requiring reinforcement learning or human feedback.

4.5 Ablation Study

Effect of cSFT. We study the effect of cSFT on training dynamics. As shown in Appendix D, cSFT leads to substantially faster convergence and significantly higher final performance across all three metrics (R@1, R@10, and MRR@10). Without cSFT, the model fails to learn meaningful identifier mappings even after 20 epochs, whereas the cSFT-equipped model rapidly acquires a well-structured semantic ID space within the first few epochs. This confirms that cSFT is essential for stabilizing training and aligning the quantized identifier space with the underlying embedding geometry.

Effect of Pseudo-Query Pretraining. Table 3 further examines the impact of the pseudo-query phase by comparing performance before and after incorporating synthetic DocT5Query supervision. Across different codebook depths (4–6) and embedding dimensions (256 and 512), the pseudo-query phase consistently improves R@1 by **+4.4 to +6.0**, with similar gains observed in R@10 and MRR@10. These improvements demonstrate that pseudo queries provide valuable coarse semantic structure, enabling the model to learn robust identifier mappings before transitioning to real-query supervision. The consistency of gains across architectures also indicates that the benefit of pseudo-query training is not sensitive to specific codebook configurations.

Takeaways. Together, these ablations show that (1) **cSFT is critical** for effective training of generative retrieval models with quantized semantic IDs, and (2) **pseudo-query pretraining reliably enhances retrieval accuracy** across all settings. These components jointly enable HHQ to learn compact yet expressive identifiers that support strong downstream retrieval performance.

5 Conclusion

We introduced Hyperspherical Householder Quantization (HHQ), a quantization framework that aligns document identifiers with the geometric structure of modern contrastive embedding spaces. By performing iterative Householder reflections on the unit hypersphere, HHQ produces compact and semantically coherent identifiers that faithfully preserve angular similarity. Combined with constrained supervised fine-tuning and tree-aware masking, HHQ enables efficient and structurally consistent docid generation.

Experiments on NQ and MS MARCO demonstrate that HHQ achieves competitive or superior retrieval performance while using significantly fewer tokens than existing generative retrieval approaches, leading to substantial reductions in decoding and inference cost. These results highlight the potential of geometrically grounded quantization methods for scalable model-based retrieval.

Future work includes extending HHQ to larger codebooks, exploring learned hierarchical priors, and integrating HHQ with large-scale generative reranking.

Limitations

Although HHQ produces compact and semantically aligned identifiers, several limitations remain.

First, similar to other quantization-based approaches, deeper codebook levels may exhibit *codeword collapse* during training, where only a subset of directions are frequently selected. This imbalance reduces the effective capacity of the quantization tree, particularly at larger depths. Addressing this issue may require explicit load-balancing objectives or diversity-promoting regularization, which we leave to future work.

Second, HHQ explicitly operates on the unit hypersphere and optimizes angular similarity, inheriting the inductive biases of contrastively trained embeddings. While cosine geometry is highly effective for many retrieval settings, it may be less suitable for datasets where semantic relevance is not well captured by angular structure, such as corpora with low document diversity or tightly coupled query–document pairs. In such cases, the hyperspherical assumption itself may limit the expressiveness of geometry-driven quantization.

Finally, as the number of quantization layers increases, most semantic information is typically captured by the early tokens. Subsequent layers tend to encode increasingly fine-grained variations, which may be dominated by noise and occasionally force semantically similar documents into different identifier paths. Better modeling of this phenomenon—such as uncertainty-aware quantization or adaptive depth control—remains an open challenge.

These limitations reflect fundamental trade-offs in embedding-based discrete representation learning and suggest promising directions for improving the robustness and generality of hyperspherical quantization methods.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen tau Yih, Sebastian Riedel, and Fabio Petroni. 2022a. Autoregressive search engines: Generating substrings as document identifiers. In *arXiv pre-print 2204.10628*.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022b. Autoregressive search engines: generating substrings as document identifiers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Xinyan Dai, Xiao Yan, Kelvin KW Ng, Jiu Liu, and James Cheng. 2020. Norm-explicit quantization: Improving vector quantization for maximum inner product search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 51–58.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

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.

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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, and 1 others. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6636–6648, Toronto, Canada. Association for Computational Linguistics.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to rank in generative retrieval. In *Proceedings of the Thirty-Eighth AAAI*

Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Kidist Amde Mekonnen, Yubao Tang, and Maarten de Rijke. 2025. Lightweight and direct document relevance optimization for generative information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 1327–1338, New York, NY, USA. Association for Computing Machinery.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to doctttttquery. *Online preprint*, 6(2).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Keshavan, Trung Vu, Lukasz Heidt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender systems with generative retrieval. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

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

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 605–611, Online. Association for Computational Linguistics.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. 2023. Learning to tokenize for generative retrieval. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A neural corpus indexer for document retrieval. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Xuefei Zhe, Shifeng Chen, and Hong Yan. 2019. Directional statistics-based deep metric learning for image classification and retrieval. *Pattern Recognition*, 93:113–123.

Yujia Zhou, Jing Yao, Zhicheng Dou, Yiteng Tu, Ledell Wu, Tat-Seng Chua, and Ji-Rong Wen. 2024a. Roger: Ranking-oriented generative retrieval. *ACM Trans. Inf. Syst.*, 42(6).

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2022. Dynamicretriever: A pre-training model-based ir system with neither sparse nor dense index. *arXiv preprint arXiv:2203.00537*.

Yujia Zhou, Jing Yao, Ledell Wu, Zhicheng Dou, and Ji-Rong Wen. 2024b. Webultron: An ultimate retriever on webpages under the model-centric paradigm. *IEEE Trans. on Knowl. and Data Eng.*, 36(9):4996–5006.

Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*.

A Dataset Statistics and Implementation Details

This appendix provides detailed statistics and construction details for the datasets used in our experiments. Table 4 summarizes the number of documents, training queries, and development queries for each dataset subset.

MS MARCO. The MS MARCO Document Ranking dataset contains approximately 3.2M web documents and 367K supervised training queries. Following WebUltron, we construct two 300K-document subsets. The *Relevant 300K* subset includes documents that have at least one labeled relevant query, while the *Random 300K* subset consists of documents randomly sampled from the full corpus. For consistency and fair comparison, we use the same randomly sampled document set as WebUltron. Queries are filtered to ensure that their relevant documents appear in the corresponding subset.

Natural Questions. We follow the NQ320K setup used in DDRO, which contains query–document pairs extracted from Wikipedia. Since multiple queries may refer to the same page, documents are deduplicated by URL. The predefined training and development splits are used without modification.

B Embedding Choice and Dimensionality Reduction

This appendix analyzes the choice of embedding model and representation dimensionality used in our experiments. Table 5 reports retrieval performance of Qwen3-Embedding-8B under different Matryoshka Representation Learning (MRL) truncation dimensions.

Recent embedding models have demonstrated strong retrieval performance across diverse benchmarks, and Qwen3-Embedding-8B represents one of the strongest open-source embedding models currently available. As shown in Table 5, the full embedding achieves strong results on both MS MARCO and Natural Questions, indicating that embedding quality is not a limiting factor in our setting.

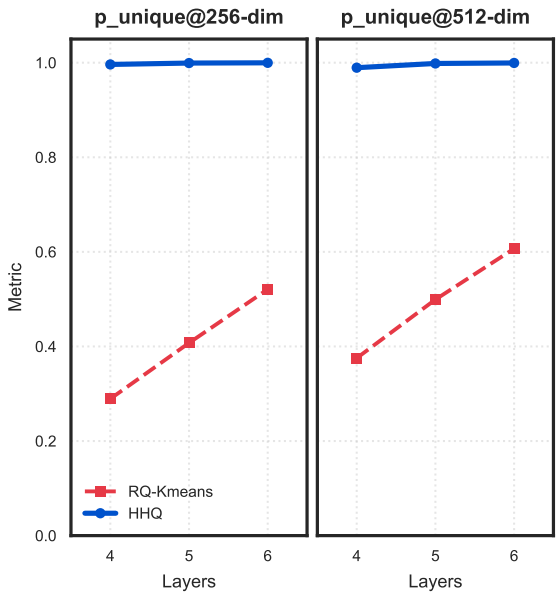

Figure 4: Comparison of p_unique across different codebook depths under 256- and 512-dim settings. Since PQ is constrained by embedding dimensionality and RQ-VAE is known to train unstably, we compare against the widely adopted RQ-KMeans baseline (e.g., in OneRec). HHQ consistently achieves near-perfect code utilization across all settings, whereas RQ-KMeans suffers from substantial unused code space.

Importantly, MRL enables embeddings to be truncated to lower dimensions while retaining most of their retrieval effectiveness. Across all datasets, reduced-dimensional representations remain competitive with the full embedding, with only modest degradation. This property allows us to substantially reduce computational and memory cost during quantization.

Based on these observations, we adopt MRL-truncated embeddings in our quantization pipeline and rely on HHQ to preserve semantic structure when distilling continuous embeddings into discrete identifiers.

C Analysis of Code Utilization

Figure 4 analyzes the code utilization behavior of different quantization strategies by reporting **p_unique**, defined as the ratio of valid unique codes to the total number of documents. This metric reflects how efficiently a quantization method uses its available code space and is particularly relevant for generative retrieval, where unused or invalid codes increase decoding ambiguity and inference cost.

We compare HHQ against RQ-KMeans, a

Dataset	#Doc	#Train Query	#Dev Query
MS MARCO Relevant 300K	319,927	367,013	808
MS MARCO Random 300K	321,631	36,670	504
NQ Relevant 320K	109,739	307,373	7,830

Table 4: Dataset statistics. The NQ dataset setup follows DDRO, and the MS MARCO dataset setup follows WebUltron. The MS MARCO Random 300K split uses the identical set of randomly sampled documents as in their experiments.

Model	MS MARCO						Natural Questions		
	Relevant 300K			Random 300K			Relevant 320K		
	R@1	R@10	MRR@10	R@1	R@10	MRR@10	R@1	R@10	MRR@10
Original									
Qwen3-Embedding-8B	34.41	81.31	49.20	58.53	91.47	70.59	59.28	89.59	70.26
MRL (Rep. Size)									
256	32.05	78.47	46.78	55.95	89.09	67.71	51.66	82.82	62.27
512	34.03	80.32	48.39	57.34	91.07	69.61	55.87	87.22	66.77
1024	33.42	80.94	48.22	57.94	91.87	70.34	57.66	88.76	68.70
2048	33.91	81.06	48.85	58.13	91.47	70.53	58.75	89.21	69.75

Table 5: Retrieval performance of Qwen3-Embedding-8B under different MRL truncation dimensions, showing that strong embedding quality is largely preserved after dimensionality reduction, which motivates our use of MRL-truncated embeddings for efficient quantization.

widely adopted residual quantization baseline used in recent generative retrieval and recommendation systems (e.g., OneRec). We do not include PQ or RQ-VAE in this comparison: PQ is fundamentally constrained by embedding dimensionality and becomes ineffective under small codebook depths, while RQ-VAE is known to suffer from training instability in large-scale settings.

As shown in Figure 4, HHQ consistently achieves near-perfect code utilization across all tested configurations, including different embedding dimensions (256 and 512) and shallow codebook depths (4–6 layers). In contrast, RQ-KMeans exhibits substantial under-utilization of the code space, with p_unique remaining well below 1 even as the number of layers increases. This gap is especially pronounced at smaller depths, where HHQ already approaches full coverage while RQ-KMeans leaves a large fraction of codes unused.

These results indicate that HHQ is significantly more effective at allocating discrete identifiers to documents, even with a compact number of tokens. High code utilization directly benefits generative retrieval by reducing identifier collisions and enabling constrained decoding with shorter sequences, thereby providing a foundation for the improved efficiency observed in our main experiments.

D Additional Ablation on cSFT

This appendix provides additional evidence on the effect of constrained supervised fine-tuning (cSFT). Figure 5 compares training dynamics on NQ320K with and without cSFT across R@1, R@10, and MRR@10.

Without cSFT, the model fails to learn meaningful semantic identifiers, showing slow convergence and consistently low retrieval performance even after extended training. In contrast, cSFT enables rapid convergence within the first few epochs and leads to substantially higher final performance across all metrics. These results support the role of cSFT in stabilizing training and enforcing alignment between the generative model and the tree-structured quantization space.

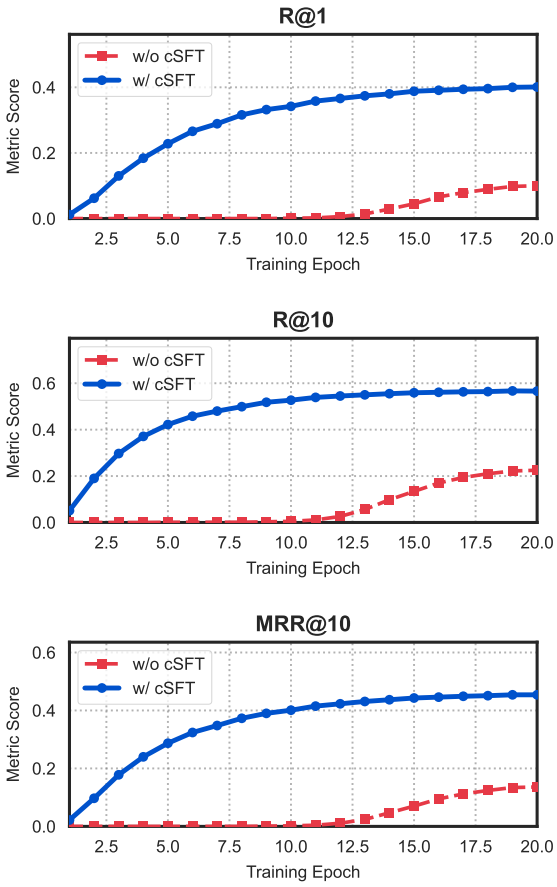

Figure 5: Ablation results on NQ320K showing that cSFT substantially accelerates training and leads to significantly higher R@1, R@10, and MRR@10 compared to training without cSFT.