

GR²: Mitigating Granularity Collapse via Adaptive Query Routing in Generative Retrieval

Soona Hong¹, Dohyeon Lee^{2,1}, Jaeyoung kim^{3*}, Seung-won Hwang^{1†}

Seoul National University¹,

Korea Advanced Institute of Science and Technology (KAIST)²

LG AI Research³

{hongsoona, seungwonh}@snu.ac.kr waylight3@kaist.ac.kr

jae.young@lgresearch.ai

Abstract

Generative Retrieval (GR) aims to directly generate document identifiers (DocIDs) without external indices. We identify **granularity collapse**: during training, GR models develop a preference for generating DocIDs with a specific granularity, causing retrieval failures when it differs from that of the target document’s DocID. We propose to quantify this mismatch using “isotropy” and confirm that retrieval accuracy degrades as such granularity mismatch increases. To address this, we propose Granularity Routing for Generative Retrieval (GR²), which trains multiple expert models specialized for different granularity levels and routes each query to the appropriate expert. Experiments on NQ320k demonstrate that GR² improves retrieval performance, with larger gains on queries that suffer more from granularity collapse.

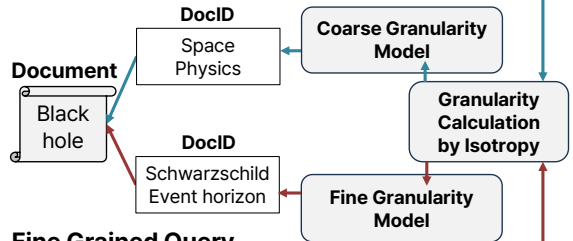
1 Introduction

Generative Retrieval (GR) aims at a model-as-index approach to address the storage and latency bottlenecks of dense retrieval (Tay et al., 2022; Bevilacqua et al., 2022). Specifically, GR encodes information directly into model parameters and eliminates the need for a separate vector index. During pretraining, GR models learn to output the Document Identifier (DocID) of a specific document when given a relevant query. These DocIDs usually consist of the document’s key terms, and retrieval performance depends heavily on how these DocIDs are constructed. Therefore, previous research has focused on creating robust DocIDs that can handle diverse queries (Lee et al., 2023; Cheng et al., 2025).

We argue that a single GR model has limited capacity to learn DocIDs due to diverse **granularity** of terms. Granularity describes the hierarchy of term specificity where superordinate terms

Coarse Grained Query

What happens in space near massive objects?



Fine Grained Query

What is the Schwarzschild radius of a stellar black hole?

Figure 1: Overview of GR² inference. Given a query, we first measure its intended granularity. Queries preferring coarse-grained DocIDs are routed to the coarse-granularity expert, which generates DocIDs using broad semantic tokens (e.g., “Space,” “Physics”). Queries preferring fine-grained DocIDs are routed to the fine-granularity expert, which generates DocIDs using specific tokens (e.g., “Schwarzschild,” “Event horizon”). This routing mitigates granularity collapse by matching query preferences to specialized experts.

are coarse-grained and subordinate terms are fine-grained (Yan et al., 2011). For example, “radius of a black hole” is broad, but “Schwarzschild radius of a stellar black hole” is specific. We find this mismatch is reflected in the DocIDs of the two documents learned by GR models.

Our experiments show that standard training causes the GR model to develop a preference for one of two granularity levels per document. We term this phenomenon **granularity collapse**. During indexing, each document is assigned a DocID with a particular granularity. Some documents receive fine-grained DocIDs (e.g., “Schwarzschild radius”) while others receive coarse-grained DocIDs (e.g., “Space Physics”). When a query is given, the model attempts to generate a DocID with its intended granularity. If this preferred granularity differs from that of the ground-truth document’s DocID, retrieval fails. For instance, if the model prefers coarse-grained DocIDs but the target docu-

* Work done while at Seoul National University.

† Corresponding Authors

ment has a fine-grained DocID, the model cannot generate the correct identifier.

We quantify this phenomenon using isotropy as a metric for granularity, motivated by prior work showing that term specificity is encoded in embedding geometry (Xypolopoulos et al., 2021; Wu and Bruch, 2025). Our analysis confirms that retrieval accuracy degrades as the granularity mismatch between query-generated and document-assigned DocIDs increases.

To address this issue, we propose **Granularity Routing for Generative Retrieval (GR²)**. Since a single model struggles to generate DocIDs across all granularities (Xu et al., 2025), GR² trains multiple expert models, each specialized for a specific granularity level. During training, each expert learns only the DocIDs corresponding to its assigned granularity (e.g., fine-grained expert, coarse-grained expert).

Figure 1 illustrates an example of how GR² uses the two expert models. Consider a document about black holes. At the training time, the coarse-granularity model assigns broad DocIDs such as “Space” and “Physics” while the fine-granularity model assigns specific DocIDs such as “Schwarzschild” and “Event horizon.” During inference, GR² first measures the query’s preferred granularity using isotropy. A coarse-grained query like “What happens in space near massive objects?” is routed to the coarse-granularity expert, whereas a fine-grained query like “What is the Schwarzschild radius of a stellar black hole?” is routed to the fine-granularity expert.

Each expert then generates the final DocID using tokens that match the query’s specificity level. This architecture is efficient because it activates only a single expert during inference and adds minimal routing overhead.

Experimental results on the NQ320k dataset show that GR² mitigates the performance degradation caused by granularity collapse and improves Recall@1 by 2.0%p. Moreover, GR² maintains inference latency comparable to or faster than existing GR models, which validates the efficiency of our routing strategy.

2 Related Work

2.1 Granularity

Queries vary in granularity from broad (e.g., *cars*) to narrow (e.g., *normocytic anemia*) (Broder, 2002), and retrieval systems should return results at the ap-

propriate level of specificity (Lalmas and Tombros, 2007). Recognizing this, granularity-aware approaches have been proposed: hierarchical methods such as DHR (Liu et al., 2021) and RAPTOR (Sarathi et al., 2024) organize documents at multiple levels, while others advocate using separate models for different query types (Zhou and Croft, 2007; Devapujula et al., 2019).

Recent work formalizes the “granularity dilemma” (Xu et al., 2025): a single model struggles to optimize for both fine-grained and coarse-grained queries simultaneously. While this challenge has been studied in dense retrieval, it remains largely unexplored in generative retrieval, where static document identifiers must serve queries spanning the entire specificity spectrum.

2.2 Document Indexing in GR

Generative retrieval assigns each document a DocID that the model learns to generate (Tay et al., 2022). Early methods such as DSI (Tay et al., 2022) and NCI (Wang et al., 2022) used static numeric identifiers, while subsequent work introduced learnable assignments (Yang et al., 2023; Liu et al., 2025; Zeng et al., 2024). Lexical approaches—DSI-QG (Zhuang et al., 2022), GLEN (Lee et al., 2023), TSGen (Zhang et al., 2024), and D2Gen (Cheng et al., 2025)—leverage pretrained language model knowledge; our method builds on this framework.

Despite these advances, DocIDs remain fixed at indexing time and cannot adapt to test query characteristics. Multiple DocID methods such as ULTRON (Zhou et al., 2022), SEAL (Bevilacqua et al., 2022), and MINDER (Li et al., 2023) assign several identifiers per document, yet they do not analyze which query properties require different DocIDs.

2.3 Isotropy

Isotropy measures how uniformly representations are distributed in the embedding space (Rudman et al., 2022). Various metrics have been proposed, including average pairwise cosine similarity (Cai et al., 2021; Ethayarajh, 2019) and covariance-based measures such as IsoScore (Rudman et al., 2022). While some work suggests that isotropy improves representation quality (Wang and Isola, 2020; Reif et al., 2019; Mu and Viswanath, 2018), others argue that anisotropy can benefit downstream tasks (Kang et al., 2025; Rudman and Eickhoff, 2024).

In dense retrieval, the findings remain mixed:

CosReg (Gao et al., 2019) improves performance by enhancing isotropy, while HIL (Kim et al., 2024) shows that both isotropic and anisotropic components contribute to generalization. Despite this growing interest, the role of isotropy in generative retrieval remains unexplored.

3 Proposed Method: GR²

To identify and address the granularity collapse problem in GR, we first propose a metric to measure granularity of DocIDs (Section 3.1). Based on this granularity metric, we describe how GR² trains expert models for target granularity (Section 3.2) and how to route queries effectively during inference (Section 3.3).

3.1 Quantifying Granularity by Isotropy

Inspired by findings that specific terms cluster tightly while general terms spread across the vector space (Xypolopoulos et al., 2021; Wu and Bruch, 2025), we propose using isotropy to measure the uniformity of embedding distributions as a proxy for granularity. Specifically, we posit that high isotropy corresponds to coarse granularity, whereas low isotropy corresponds to fine granularity.

3.1.1 Definition of Granularity Score

A primary challenge in applying isotropy directly to DocID embeddings is the limited sample size. Since DocIDs typically consist of only a few tokens, measuring isotropy from a single DocID is unreliable due to high estimation variance (Rudman and Eickhoff, 2024).

To address this challenge, we generate multiple DocIDs via beam search and measure the isotropy of their combined token embeddings. For a query q , we perform constrained beam search to generate DocIDs from the existing document index, while we use unconstrained beam search for a document d . In both cases, we retain the top- K candidates instead of selecting only the single most probable sequence. Let $e_i^t \in \mathbb{R}^D$ denote the embedding of the t -th token in the i -th DocID candidate, where D is the embedding dimension and each DocID consists of T tokens. We collect all token embeddings:

$$\mathcal{E}_d = \{e_i^t : 1 \leq i \leq K, 1 \leq t \leq T\} \quad (1)$$

Subsequently, we compute the granularity score $S(d)$ based on the isotropy of these $K \times T$ token embeddings using IsoScore (Rudman et al., 2022):

$$S(d) = \text{IsoScore}(\mathcal{E}_d) \quad (2)$$

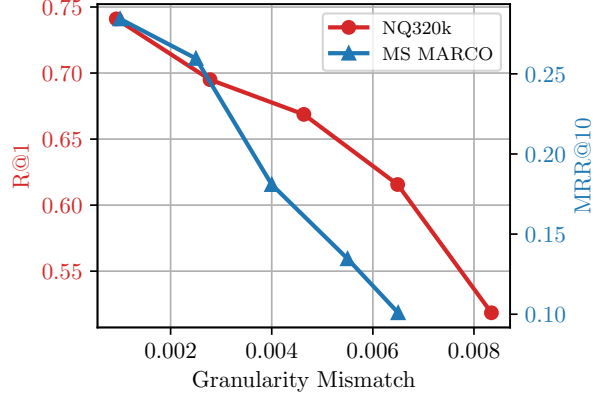


Figure 2: Performance decreases monotonically as the granularity mismatch increases on both NQ320k (Recall@1) and MS MARCO (MRR@10), confirming that granularity collapse degrades retrieval performance.

A score near 1 denotes high isotropy (coarse granularity), while a score near 0 denotes high anisotropy (fine granularity). Similarly, for a query q , we compute $S(q)$ by performing constrained beam search over the existing document index to generate top- K valid DocID candidates and collecting the embeddings of all $K \times T$ tokens from these candidates. Note that $S(q)$ uses constrained decoding over the existing document index, while $S(d)$ uses unconstrained decoding because document-side generation constructs the index itself, with no prior index to constrain against.

3.1.2 Granularity Collapse in GR

Based on the granularity score, we formulate the granularity mismatch $G(q, d)$ to measure the discrepancy between a query q and its ground-truth document d .

$$G(q, d) = |(S(q) - \bar{S}(q)) - (S(d) - \bar{S}(d))| \quad (3)$$

where $\bar{S}(q)$ and $\bar{S}(d)$ denote the mean granularity scores over all queries and documents, respectively.

Figure 2 illustrates the retrieval performance of GLEN on NQ320k (Kwiatkowski et al., 2019) and MS MARCO (Bajaj et al., 2016) with respect to this granularity mismatch. The results demonstrate a monotonic decrease in both datasets as the mismatch increases, indicating that even strong GR models struggle when the granularities of the query and document do not align. This implies that during training, GR models implicitly overfit to a single granularity, as each document is assigned only one DocID that encodes a single granularity. We term this phenomenon **granularity collapse**, as the model’s capability to handle varying granularities

is lost.

3.2 Training Expert Models for Target Granularity

The empirical results in Section 3.1 demonstrate that granularity mismatch directly degrades retrieval performance. Since a single model struggles to generate DocIDs across all granularities, we train multiple expert models, each specialized for a specific granularity level. To train such experts, we need a method to generate DocIDs with controlled granularity levels.

3.2.1 DocID Sampling with Target Granularity

Following Mu and Viswanath (2018), we manipulate the token embedding space via principal component analysis (PCA) to control the granularity of generated DocIDs.

Using a pretrained indexing model, we generate a preliminary DocID for each document d in the document collection $\mathcal{D}$ via beam search. We construct matrix $E \in \mathbb{R}^{|\mathcal{D}| \times T \times D}$ by collecting all token embeddings from these DocIDs with duplicates, where each row corresponds to one token embedding. Applying singular value decomposition (SVD) to the centered embedding matrix:

$$E^T = U \cdot \Sigma \cdot V^T \quad (4)$$

The columns of U are the principal components, ordered by decreasing variance.

As shown in Figure 3(a), we control granularity by scaling the principal components of the vocabulary embedding matrix $E_v \in \mathbb{R}^{|V| \times D}$, where $|V|$ is the vocabulary size. Define the scaling matrix:

$$\Lambda = \text{Diag}(\underbrace{R, R, \dots, R}_{L \text{ terms}}, \underbrace{1, 1, \dots, 1}_{D-L \text{ terms}}) \quad (5)$$

where R is the scaling factor and L is the number of top principal components to scale. The transformed embedding matrix is:

$$E'_v = E_v \cdot (U \Lambda U^T) \quad (6)$$

When $R < 1$, we compress the top principal components, making embeddings more uniformly distributed and yielding coarse-grained DocIDs. Conversely, when $R > 1$, we expand the top principal components, making embeddings more concentrated and yielding fine-grained DocIDs. We then perform beam search in the transformed embedding space to generate K DocID candidates per document.

3.2.2 DocID Selection

DocID sampling generates K DocID candidates per document with controlled granularity. However, this process treats each document independently, so a sampled DocID may be more closely associated with other documents than with its intended target.

This issue becomes critical due to the constrained decoding mechanism in GR. During retrieval, the model generates tokens constrained to existing DocIDs in the index. Even when the query and ground-truth document remain the same, changing other documents' DocIDs can alter retrieval results. For example, as illustrated in Figure 3(b), suppose a query q with ground-truth document d generates DocIDs with probabilities in the order $z_1 > z_2 > z_3$, and d is assigned z_2 as its DocID. If only z_2 and z_3 exist in the index, d would rank first. However, if another document d' is assigned z_1 as its DocID, d is pushed to rank 2, causing retrieval failure. Thus, selecting a DocID for one document depends on the DocID choices of other documents.

To address this, we simulate the retrieval process on training queries and select DocIDs that maximize overall retrieval performance. By considering competition among DocIDs during selection, we can avoid assigning DocIDs that would block each other at inference time. We use Recall@1 as the optimization target, as it directly reflects competition among DocIDs: if another document's DocID ranks higher than the ground-truth, retrieval fails.

Relaxing to continuous optimization Selecting one DocID from K candidates for each of $|\mathcal{D}|$ documents requires searching over $K^{|\mathcal{D}|}$ possible assignments. To make this tractable, we relax the discrete selection to a continuous optimization problem. We introduce a learnable score $A_{d,j} \in \mathbb{R}$ for each candidate z_d^j (the j -th candidate for document d), and define the selection probability as:

$$P(d \rightarrow z_d^j) = \frac{\exp(A_{d,j}/\tau)}{\sum_{k=1}^K \exp(A_{d,k}/\tau)} \quad (7)$$

where τ is a temperature parameter.

Differentiable approximation of Recall@1 Recall@1 equals 1 if the ground-truth document is retrieved at rank 1, and 0 otherwise. To approximate this with a differentiable function, we compute the probability that the ground-truth is retrieved first.

For each query q with ground-truth document d_q , beam search yields a ranked list $L_q =$

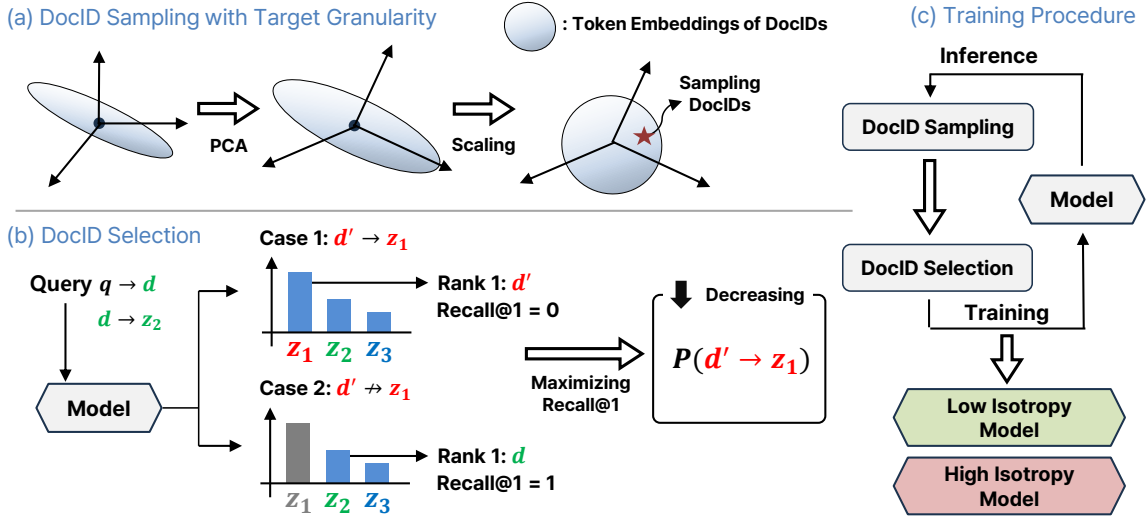


Figure 3: Overview of GR^2 . (a) **DocID Sampling**: We apply PCA to token embeddings to identify principal components, then scale along these directions to control granularity and sample DocIDs with target characteristics. (b) **DocID Selection**: For a query q with ground-truth d where $d \rightarrow z_2$, the model generates DocIDs with probabilities $z_1 > z_2 > z_3$. If another document d' is assigned z_1 (Case 1), d' ranks first and $\text{Recall}@1 = 0$. If d' is not assigned z_1 (Case 2), d ranks first and $\text{Recall}@1 = 1$. To maximize $\text{Recall}@1$, we decrease $P(d' \rightarrow z_1)$. (c) **Training Procedure**: Each training epoch iterates through DocID Sampling, DocID Selection, and Model Training. At inference, queries are routed to either the fine-granularity or coarse-granularity expert based on their preferred granularity.

$[z_{d_1}^{j_1}, z_{d_2}^{j_2}, \dots]$ where d_i is the document at rank i . For the ground-truth to be retrieved at some rank i , three conditions must hold: (1) the ground-truth document appears at rank i , (2) the candidate at rank i is selected as its document's DocID, and (3) all candidates at higher ranks are not selected, so they do not block rank i . Let $w_{q,i}$ denote the probability that no higher-ranked candidate is selected. Since $(1 - P(d_k \rightarrow z_{d_k}^{j_k}))$ is the probability that the candidate at rank k is not selected, their product gives:

$$w_{q,i} = \prod_{k=1}^{i-1} (1 - P(d_k \rightarrow z_{d_k}^{j_k})) \quad (8)$$

Multiplying the probabilities of these three conditions and summing over all ranks gives:

$$V_q = \sum_{i=1}^{|L_q|} \mathbf{1}[d_i = d_q] \cdot P(d_i \rightarrow z_{d_i}^{j_i}) \cdot w_{q,i} \quad (9)$$

where $\mathbf{1}[d_i = d_q]$ is the indicator for condition (1), $P(d_i \rightarrow z_{d_i}^{j_i})$ is the probability for condition (2), and $w_{q,i}$ is the probability for condition (3).

Loss function We maximize retrieval probability across training queries Q :

$$\mathcal{L} = -\frac{1}{|Q|} \sum_{q \in Q} \log V_q \quad (10)$$

Maximizing V_q has two effects: increasing the selection probability for ground-truth candidates, and decreasing it for other candidates that would block the ground-truth at higher ranks. After optimization, the final DocID for document d is $z_d^* = z_d^{j_d^*}$ where $j^* = \arg\max_j A_{d,j}$.

3.2.3 Model Training

Using the selected DocIDs from the previous step, we train the retrieval model via supervised learning. The model learns to generate the selected DocID when given a relevant query.

As training progresses, the model's behavior changes, which in turn affects which DocIDs are optimal. As illustrated in Figure 3(c), we repeat DocID Sampling, DocID Selection, and Model Training at each epoch, continuously updating the DocIDs to match the current model state.

3.3 Inference with Granularity Routing

At inference time, we route each query to the expert model that best matches its preferred granularity. The fine-granularity expert first generates a preliminary DocID via constrained beam search. We then compute the query's granularity score S_q from the top- K candidates using the method described in Section 3.1. Based on a threshold hyperparameter γ , we route the query to the appropriate expert:

queries with $S_q > \gamma$ (preferring coarse-grained DocIDs) are processed by the coarse-granularity expert, while the rest are processed by the fine-granularity expert. This routing mitigates granularity collapse by matching query preferences to specialized experts, and activates only a single model during inference, adding minimal overhead. Since each expert assigns a DocID of different granularity to the same document, this enables queries with different granularities to retrieve the same document, each through the expert with a matching DocID.

4 Experiments

4.1 Experimental Settings

4.1.1 Dataset

We evaluate our approach on NQ320k, a widely used dataset in generative retrieval, and test generalization on TriviaQA. (1) NQ320k (Kwiatkowski et al., 2019) is a dataset for evaluating retrieval performance in knowledge-intensive QA settings, consisting of 109k documents, 320k training queries, and 7,830 test queries. (2) TriviaQA (Joshi et al., 2017) is a reading comprehension dataset consisting of trivia questions. We evaluate TriviaQA test queries on the NQ320k corpus, using only queries whose answer documents exist in the corpus, resulting in 5,210 test queries.

4.1.2 Evaluation Metric

We evaluate retrieval performance using Recall@ k (whether a relevant document appears in the top- k) and MRR@ k (reciprocal rank of the first relevant document), following existing GR benchmarks.

4.1.3 Baselines

We compare GR² with recent generative retrieval methods. GLEN (Lee et al., 2023) learns DocIDs through lexical index learning and serves as our primary backbone model. SEAL (Bevilacqua et al., 2022) and MINDER (Li et al., 2023) assign multiple DocIDs per document to handle diverse queries. TSGen (Zhang et al., 2024) generates term sets without imposing a fixed order. D2Gen (Cheng et al., 2025) constructs DocIDs that are both descriptive and discriminative, achieving strong performance.¹ NCI (Wang et al., 2022) uses hierarchical numeric identifiers with a prefix-aware decoder. Since no pretrained checkpoint with T5-base as the

backbone is publicly available, we train NCI on NQ320k using the released codebase.

4.1.4 Implementation Details

We use GLEN (Lee et al., 2023) as the primary backbone model for GR², since it uses a single model for both indexing and retrieval, which simplifies interpretation. We train for 20 epochs with learning rate 1×10^{-4} . For granularity measurement, we generate $K = 8$ candidate DocIDs per document and scale the top $L = 8$ principal components following Mu and Viswanath (2018). We set $R = 0.1$ for the coarse-granularity expert and $R = 1.1$ for the fine-granularity expert based on Recall@1 on training queries (Appendix A.1).

For inference-time routing between experts, we set the threshold γ to the median granularity score of NQ320k test queries. Appendix A.2 shows that performance is stable across different thresholds. The two expert models are initialized from the same pretrained backbone and trained independently with granularity-specific DocIDs.

The core components of GR² require only access to DocID token embeddings and constrained decoding over valid identifiers. To verify that GR² is not tied to a single architecture, we additionally apply it to NCI, which uses a different type of DocID from GLEN.

4.2 Results and Analysis

In Section 3.1, we showed that granularity mismatch between queries and DocIDs degrades retrieval performance. We now evaluate whether GR² successfully addresses this problem through the following research questions:

- **RQ1 (Retrieval Performance):** Does GR² improve retrieval performance over existing methods?
- **RQ2 (Effect on Granularity Mismatch):** Does GR² reduce performance degradation caused by granularity mismatch?
- **RQ3 (Ablation Study):** How does each component of GR² contribute to mitigating granularity mismatch?
- **RQ4 (Efficiency and Generalization):** Is GR² efficient and does it generalize to different query distributions?

¹As the D2Gen model is not publicly available, we report results from the original paper and limit comparison to the main evaluation.

Method	M@10	M@100	R@1	R@10	R@100
SEAL	-	0.677	0.599	0.812	0.909
MINDER	0.709	0.731	0.627	0.869	0.933
TSGen	0.771	0.774	0.708	0.889	0.948
D2Gen	<u>0.772</u>	0.774	<u>0.710</u>	0.876	0.936
NCI	0.711	0.729	0.657	0.850	0.922
+ GR ²	0.714	0.731	0.668	0.855	0.924
GLEN	0.751	<u>0.754</u>	0.692	0.860	0.921
+ GR ²	0.773	0.774	0.712	<u>0.879</u>	<u>0.938</u>

Table 1: Retrieval performance on NQ320k. Baseline results are from Cheng et al. (2025), except for NCI, which we reproduce using the released codebase. M and R indicate MRR and Recall, respectively. **Bold** and underline indicate the best and second-best results, respectively.

Model	Average Document Granularity Score (%)
GLEN (Baseline)	0.334
GR ² (Coarse Granularity)	0.353
GR ² (Fine Granularity)	0.325

Table 2: Average granularity score of document on NQ320k. GR² successfully controls document granularity to target levels.

4.2.1 RQ1: Retrieval Performance

Table 1 presents the retrieval performance on NQ320k. GR² improves R@1 from 69.2% (GLEN) to 71.2% (+2.0%p), outperforming D2Gen (71.0%), the previous best R@1 among lexical DocID methods. The improvement is consistent across other metrics: M@10 increases from 0.751 to 0.773, and R@10 improves from 86.0% to 87.9%. This indicates that optimizing Recall@1 in DocID Selection (Section 3.2.2) also improves retrieval at deeper ranks. We next investigate whether this improvement concentrates on the queries most affected by granularity collapse. The improvement is not specific to GLEN. Table 1 also shows that applying GR² to NCI improves R@1 from 65.7% to 66.8% (+1.1%p), indicating that GR² is not tied to a single architecture.

4.2.2 RQ2: Effect on Granularity Mismatch

We analyze whether this improvement stems from mitigating the granularity mismatch identified in Section 3.1. For GR² to mitigate granularity mismatch, two conditions must hold: (1) the expert models must learn DocIDs with distinct granularity levels, and (2) routing queries to the appropriate expert must reduce performance degradation under

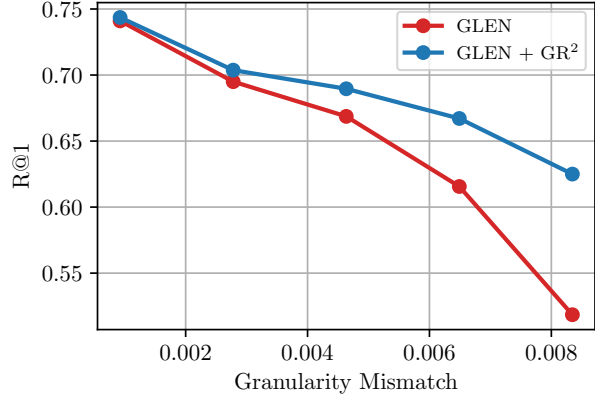


Figure 4: Recall@1 performance per granularity mismatch on NQ320k. Granularity mismatch is computed using the baseline GLEN model for both methods to ensure consistent comparison, and the same x-axis bins are used for both GLEN and GLEN + GR².

high mismatch. We verify both conditions.

Table 2 shows the average granularity score of DocIDs produced by each expert. The coarse-granularity expert ($R = 0.1$) produces DocIDs with higher granularity score (0.353%), indicating broader terms, while the fine-granularity expert ($R = 1.1$) produces DocIDs with lower granularity score (0.325%), indicating more specific terms. This confirms that DocID sampling with PCA scaling successfully creates experts specialized for different granularity levels.

Given that experts have distinct granularity characteristics, we examine whether routing reduces the performance degradation observed in Section 3.1. Figure 4 compares Recall@1 across different levels of granularity mismatch. GLEN exhibits severe degradation as mismatch increases: Recall@1 drops from 74% to 52% (22%p decline). In contrast, GR² shows a smaller decline from 74% to 63% (11%p), reducing the performance gap by 11%p. This demonstrates that routing queries to experts with appropriate granularity mitigates the negative impact of granularity mismatch. These findings suggest that the performance gains of GR² are attributable to addressing granularity collapse.

To qualitatively validate the granularity score defined in Section 3.1, Table 3 presents DocIDs from football-related documents ordered by their granularity scores. DocIDs with lower scores contain terms that refer to specific roles or entities (e.g., “captain,” “quarterback”), while DocIDs with higher scores contain broader categorical terms (e.g., “cup,” “players”). All examples share the token “football,” making the granularity contrast at-

Granularity Score (%)	DocID Tokens
0.233	captain, football, premier
0.249	quarterback, wins, football
0.250	cricket, football, dual
0.316	football, association, sanctions
0.409	players, football, pro
0.410	cup, league, football

Table 3: DocID examples from football-related documents on NQ320k, ordered by granularity score. Lower scores correspond to specific terms, while higher scores correspond to broader terms.

Model	R@1
GLEN (Baseline)	0.692
GLEN Ensemble ($\times 2$)	0.695
GR ²	0.712
w/o DocID Sampling	0.698
w/o DocID Selection	0.671
Coarse Granularity Only	0.697
Fine Granularity Only	<u>0.705</u>

Table 4: Ablation study on NQ320k. **Bold** and underline indicate the best and second-best results, respectively.

tributable to the remaining tokens. This aligns with the expectation that specific terms cluster tightly (low isotropy) while general terms spread broadly (high isotropy) (Xypolopoulos et al., 2021; Wu and Bruch, 2025).

4.2.3 RQ3: Ablation Study

Table 4 ablates each component of GR² on NQ320k. Removing DocID sampling reduces R@1 to 0.698 (-1.4%p). Without sampling, both experts learn the same DocIDs and become identical models, so this result reflects the contribution of DocID selection alone. Removing DocID selection causes a more severe degradation to R@1 of 0.671, falling below even the GLEN baseline. This shows that controlling granularity alone is insufficient; optimizing DocID assignments based on retrieval performance is essential.

Using only the coarse-granularity expert achieves 69.7%, while using only the fine-granularity expert achieves 70.5%. The coarse-granularity expert achieves lower performance because broader terms are less discriminative across documents. However, combining both experts through routing substantially outperforms either expert alone, demonstrating that the two experts

Model	Latency (s)	Recall@1
SEAL	2.45	0.599
MINDER	2.44	0.627
TSGen	1.60	<u>0.708</u>
GLEN	0.34	0.692
+ GR ²	<u>0.53</u>	0.712

Table 5: Inference latency comparison on NQ320k. The best and the second-best performances are marked in bold and underlined, respectively.

Method	Seen		Unseen	
	R@1	M@10	R@1	M@10
GLEN	0.726	0.783	0.576	0.638
+ GR ²	0.745	0.798	0.599	0.654

Table 6: Retrieval performance on NQ320k seen and unseen queries. **Bold** indicates the best result.

are complementary.

To rule out gains may come simply from the increased model capacity, we compare against a GLEN ensemble that averages log probabilities from two independently trained GLEN models. The ensemble achieves R@1 of 0.695, only marginally above single GLEN (0.692), while each individual expert in GR² (0.697 and 0.705) outperforms the ensemble and the full GR² (0.712) further improves through routing. This confirms that the gains of GR² stem from specializing experts for different granularity, not from doubling parameters.

4.2.4 RQ4: Efficiency and Generalization

Since GR² employs multiple expert models, a natural concern is whether this introduces significant computational overhead. Table 5 presents the inference latency comparison on NQ320k. GR² increases inference time from 0.34s to 0.53s per query batch, due to computing the granularity score and routing. Despite this overhead, GR² remains faster than multi-DocID approaches such as SEAL and MINDER, which require over 2.4 seconds. This efficiency stems from our routing strategy, which activates only a single expert per query. Analysis for training cost is provided in Appendix A.3.

We next examine whether the improvements generalize beyond the training distribution. Table 6 presents the retrieval performance on seen and unseen queries in NQ320k, where unseen queries tar-

Model	R@1	M@10
SEAL	<u>0.350</u>	<u>0.424</u>
MINDER	0.348	0.419
TSGen	0.288	0.352
GLEN	0.303	0.364
+ GR ²	0.363	0.427

Table 7: Generalization performance on TriviaQA using models trained on NQ320k. **Bold** and underline indicate the best and second-best results, respectively.

get documents that do not appear as ground-truth answers in training. On seen queries, GR² improves R@1 from 72.6% to 74.5% (+1.9%p). On unseen queries, GR² achieves a larger improvement from 57.6% to 59.9% (+2.3%p). The larger gain on unseen queries suggests that routing enhances generalization rather than simply memorizing training patterns.

Table 7 shows the generalization performance on TriviaQA, where queries are trivia questions rather than the search log queries of NQ320k. We evaluate models trained on NQ320k using TriviaQA queries whose answer documents exist in the NQ320k corpus. GR² achieves R@1 of 36.3%, outperforming GLEN (30.3%) by 6.0%p. This improvement on out-of-domain queries demonstrates that the routing mechanism captures general properties of query granularity rather than dataset-specific patterns.

5 Conclusion

We identified granularity collapse in generative retrieval: models develop a preference for specific granularity levels during training, leading to retrieval failures when this preference mismatches the target document’s DocID granularity. To address this, we proposed GR², which trains granularity-specific expert models and routes queries to the appropriate expert based on their preferred granularity. GR² targets the specific failure mode of granularity collapse, with the largest gains where existing models degrade the most: an 11%p reduction in mismatch-driven degradation and 6.0%p on out-of-domain queries.

Acknowledgements

This work was supported by the InnoCORE program of the Ministry of Science and ICT (N10250156), and by the MSIT(Ministry of Science and ICT), Korea, under the

ITRC(Information Technology Research Center) support program(IITP-2026-RS-2020-II201789) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

Limitations

GR² requires training and storing multiple expert models, which increases memory requirements compared to single-model approaches. Additionally, the inference process involves a preliminary decoding step to measure query granularity before routing, which adds latency overhead compared to direct generation. Furthermore, our granularity measurement relies on the isotropy of lexical token embeddings. We show that larger granularity mismatch correlates with lower retrieval performance. However, the score has not been directly validated against human judgments of semantic specificity. Such annotation-based validation would further strengthen the empirical grounding of the score. While we validate GR² on two architectures with different identifier designs, applying it to a broader range of generative retrieval methods remains future work. Finally, GR² currently uses a binary split into coarse and fine granularity; extending to more granularity levels may further reduce mismatch but increases training and routing complexity.

References

- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. [MS MARCO: A human generated machine reading comprehension dataset](#). *arXiv preprint arXiv:1611.09268*.
- Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.
- Andrei Broder. 2002. [A taxonomy of web search](#). *SIGIR Forum*, 36(2):3–10.
- Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International conference on learning representations*.
- Jiehan Cheng, Zhicheng Dou, Yutao Zhu, and Xiaoxi Li. 2025. [Descriptive and discriminative document identifiers for generative retrieval](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(11):11518–11526.

- Siddhartha Devapujula, Sagar Arora, and Sumit Borar. 2019. [Learning to rank broad and narrow queries in e-commerce](#). *Preprint*, arXiv:1907.01549.
- 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.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. [Representation degeneration problem in training natural language generation models](#). In *International Conference on Learning Representations*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Seil Kang, Woojung Han, Dayun Ju, and Seong Jae Hwang. 2025. [Rare text semantics were always there in your diffusion transformer](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Jaeyoung Kim, Dohyeon Lee, and Seung-won Hwang. 2024. [HIL: Hybrid isotropy learning for zero-shot performance in dense retrieval](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7892–7903, Mexico City, Mexico. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Mounia Lalmas and Anastasios Tombros. 2007. [Evaluating xml retrieval effectiveness at inex](#). *SIGIR Forum*, 41(1):40–57.
- Sunkyoung Lee, Minjin Choi, and Jongwuk Lee. 2023. [GLEN: Generative retrieval via lexical index learning](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7693–7704, Singapore. Association for Computational Linguistics.
- 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.
- Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. 2021. [Dense hierarchical retrieval for open-domain question answering](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yuxuan Liu, Tianchi Yang, Zihan Zhang, Minghui Song, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. 2025. [Towards distributionally balanced end-to-end generative retrieval](#). In *Companion Proceedings of the ACM on Web Conference 2025, WWW '25*, page 384–393, New York, NY, USA. Association for Computing Machinery.
- Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective post-processing for word representations. In *6th International Conference on Learning Representations, ICLR 2018*.
- Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. *Advances in neural information processing systems*, 32.
- William Rudman and Carsten Eickhoff. 2024. [Stable anisotropic regularization](#). In *The Twelfth International Conference on Learning Representations*.
- William Rudman, Nate Gillman, Taylor Rayne, and Carsten Eickhoff. 2022. [IsoScore: Measuring the uniformity of embedding space utilization](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3325–3339, Dublin, Ireland. Association for Computational Linguistics.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. [RAPTOR: Recursive abstractive processing for tree-organized retrieval](#). In *The Twelfth International Conference on Learning Representations*.
- Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, and 1 others. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.
- Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.
- Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, and 1 others. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

Si Wu and Sebastian Bruch. 2025. [Uncovering visual-semantic psycholinguistic properties from the distributional structure of text embedding space](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5638–5649, Vienna, Austria. Association for Computational Linguistics.

Liyan Xu, Zhenlin Su, Mo Yu, Jiangnan Li, Fandong Meng, and Jie Zhou. 2025. [Dense retrievers can fail on simple queries: Revealing the granularity dilemma of embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19295–19305, Suzhou, China. Association for Computational Linguistics.

Christos Xypolopoulos, Antoine Tixier, and Michalis Vazirgiannis. 2021. [Unsupervised word polysemy quantification with multiresolution grids of contextual embeddings](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3391–3401, Online. Association for Computational Linguistics.

Xin Yan, Raymond Y.K. Lau, Dawei Song, Xue Li, and Jian Ma. 2011. [Toward a semantic granularity model for domain-specific information retrieval](#). *ACM Trans. Inf. Syst.*, 29(3).

Tianchi Yang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. [Auto search indexer for end-to-end document retrieval](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6955–6970, Singapore. Association for Computational Linguistics.

Hansi Zeng, Chen Luo, and Hamed Zamani. 2024. [Planning ahead in generative retrieval: Guiding autoregressive generation through simultaneous decoding](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’24*, page 469–480, New York, NY, USA. Association for Computing Machinery.

Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. 2024. [Generative retrieval via term set generation](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 458–468.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. [Ultron: An ultimate retriever on corpus with a model-based indexer](#). *arXiv preprint arXiv:2208.09257*.

Yun Zhou and W. Bruce Croft. 2007. [Query performance prediction in web search environments](#). In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’07*, page 543–550, New York, NY, USA. Association for Computing Machinery.

Learning Rate	Training Loss
1×10^{-4}	0.9704
3×10^{-4}	0.9822
5×10^{-4}	0.9954

Table 8: Learning rate selection based on training loss. We select 1×10^{-4} .

Scaling Factor	R@1	
	Coarse-Granularity	Fine-Granularity
0.1	0.9087	–
0.3	0.9068	–
0.5	0.9073	–
0.7	0.9062	–
1.1	–	0.9065
1.3	–	0.9012
1.5	–	0.8595
1.7	–	0.8294

Table 9: Scaling factor selection based on Recall@1 on training queries. We select $R = 0.1$ for the coarse-granularity expert and $R = 1.1$ for the fine-granularity expert.

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 Appendix

A.1 Hyperparameter Selection

We select hyperparameters based on training set performance. For the learning rate, we search over $\{1, 3, 5\} \times 10^{-4}$ and select the value that achieves the lowest training loss. For the scaling factor R , we search over $\{0.1, 0.3, 0.5, 0.7\}$ for the coarse-granularity expert and $\{1.1, 1.3, 1.5, 1.7\}$ for the fine-granularity expert, selecting the value that achieves the highest Recall@1 on training queries.

Table 8 and Table 9 report the results of hyperparameter selection on NQ320k.

A.2 Sensitivity of Threshold γ

We examine the sensitivity of GR^2 to different threshold values. Table 10 reports R@1 when γ is set to different percentiles of the test granularity scores. Performance varies only marginally across the tested thresholds, and GR^2 consistently outperforms GLEN in all settings. The 75th-percentile threshold also slightly outperforms the reported median setting, showing that the improvement is

γ (percentile of test $S(q)$)	R@1
25th	0.703
50th	0.712
75th	0.713

Table 10: Sensitivity of GR² to the routing threshold γ on GLEN and NQ320k.

Component	Time	Ratio (%)
DocID Sampling	1m 16s	5.6
DocID Selection	3m 44s	16.4
Model Training	17m 45s	78.0
Total (per epoch)	22m 45s	100.0

Table 11: Wall-clock training time per epoch on NQ320k (NVIDIA 3090 $\times$ 4).

robust to threshold selection and is not driven by choosing the best-performing value.

A.3 Training Cost

Table 11 presents the wall-clock training time per epoch. DocID Sampling and Selection add 5 minutes to each epoch, accounting for 22% of the total. Since GR² trains two experts, the total training time is approximately twice that of GLEN, though the two experts can be trained in parallel.

A.4 Training Algorithm

Algorithm 1 summarizes the complete training procedure of GR², which iterates through DocID Sampling, DocID Selection, and Model Training at each epoch.

A.5 Licenses and Intended Use

All artifacts used in this work are publicly available and used in accordance with their intended purposes. Natural Questions (Kwiatkowski et al., 2019), from which NQ320k is derived, is released under the CC BY-SA 3.0 license. TriviaQA (Joshi et al., 2017) is released under the Apache License 2.0. The GLEN codebase (Lee et al., 2023), which serves as our backbone implementation, is released under the Apache License 2.0. All datasets and code are used solely for academic research, consistent with their original access conditions.

Algorithm 1 GR² Training Process

```

1: Input: Document set  $\mathcal{D}$ , Query set  $\mathcal{Q}$ , Scaling factor  $R$ , Epochs  $E$ 
2: Output: Trained model  $M$  with optimized DocIDs
3: Initialize model parameters  $\theta$  and DocID selection parameters  $A$ 
4: for epoch = 1 to  $E$  do
5:   // Step 1: DocID Sampling
6:   Compute embedding matrix  $E$  from current model
7:   Apply SVD:  $E^T = U \cdot \Sigma \cdot V^T$ 
8:   Transform embeddings:  $E'_v = E_v \cdot (U \Lambda U^T)$  with  $\Lambda$  using  $R$ 
9:   Generate  $K$  DocID candidates per document using beam search with  $E'_v$ 
10:  // Step 2: DocID Selection
11:  for each mini-batch of queries do
12:    Perform constrained decoding with current DocIDs
13:    Update selection parameters  $A$  to minimize loss  $\mathcal{L}$ 
14:  end for
15:  Select final DocID for each document based on updated  $A$ 
16:  // Step 3: Model Training
17:  Train model  $M$  for one epoch using selected DocIDs
18:  Update model parameters  $\theta$  via gradient descent
19: end for
20: return Trained model  $M$  with final DocIDs

```
