

Training-Free Generative Retrieval

Yingchen Zhang^{1,2,3} Ruqing Zhang^{1,3*} Jiafeng Guo^{1,3}
Maarten de Rijke⁴ Fuyu Lv⁵ Xueqi Cheng^{1,3}

¹ State Key Laboratory of AI Safety

² School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences

³ Institute of Computing Technology, Chinese Academy of Sciences

⁴ University of Amsterdam

⁵ Taobao & Tmall Group of Alibaba

{zhangyingchen23s, zhangruqing, guojiafeng}@ict.ac.cn,
m.derijke@uva.nl, fuyu.lfy@alibaba-inc.com, cxq@ict.ac.cn

Abstract

Large language models (LLMs) are emerging as versatile foundations for generative retrieval (GR), a paradigm that directly generates relevant document identifiers (docids) given a query. Existing ways of using LLMs for GR typically rely on heavy post-hoc training to fit a relevance data prior, often leading to overfitting and poor cross-domain generalization. Can retrieval, like other language tasks, be performed without retriever-specific training using LLMs? We conduct a preliminary study and find that constrained decoding, required to ensure valid docids, often misaligns with the model’s natural free-generation patterns. We propose FreeGR, a plug-and-play framework that adapts retrieval data to a LLM’s habitual generation behavior. FreeGR consists of three components: an LLM self-annotated docid generator, an LLM-preferred input context designer, and two decoding strategies that trade off retrieval effectiveness and efficiency. It requires no retriever-specific parameter optimization, is model-agnostic, and scales across LLM families and sizes while preserving their generative capabilities. Experiments on the NQ, MS MARCO, and BEIR datasets show that FreeGR achieves competitive retrieval effectiveness compared with several trained generative and dense retrievers, without retriever-specific parameter optimization.

1 Introduction

Generative retrieval (GR) has emerged as a novel information retrieval paradigm, which directly generates identifiers (docids) of relevant documents for a given query (Tay et al., 2022; Wang et al., 2022b).

Heavy post-hoc training to fit the data distribution. Large language models (LLMs) (Lewis et al., 2020; Raffel et al., 2019) provide a natural foundation for GR because of their strong generation ability and flexible input handling. As shown in

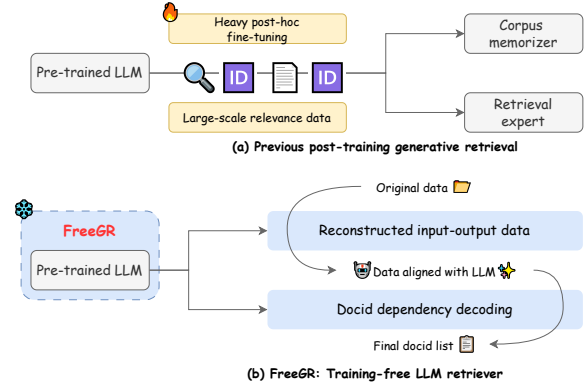


Figure 1: Comparison between prior post-training generative retrievers and our FreeGR. (a) GR methods usually depend on heavy data curation and fine-tuning; (b) FreeGR directly uses LLMs as generative retrievers without retriever-specific parameter optimization.

Figure 1(a), most approaches to putting LLMs to work for GR adapt LLMs to GR through heavy post-hoc fine-tuning (Tay et al., 2022; Yang et al., 2023; Wang et al., 2023). We refer to this paradigm as *model-fits-data*: the LLM is further trained to approximate the empirical data-prior relevance mapping from queries to docids. This approach depends critically on large quantities of high-quality paired data (Zhuang et al., 2022).

However, the model-fits-data paradigm faces two intertwined challenges. (i) In practice, collecting large-scale and unbiased relevance data is difficult, leading the model to overfit the limited supervision available. The model then degenerates into memorizing docids associated with a narrow set of training queries (Wang et al., 2022b; Pradeep et al., 2023). (ii) Such rote memorization distorts the LLM’s inherent language understanding and generation abilities, resulting in poor performance on unseen data and weak generalization (Sun et al., 2025) (see Appendix D for empirical evidence).

Recent studies such as Few-Shot GR and ZeroGR suggest that pretrained LLMs already encode substantial relevance knowledge acquired during pre-training, which can serve as a strong

* Corresponding author

prior for retrieval (Askari et al., 2024; Sun et al., 2025). However, existing training-free attempts remain limited. Few-Shot GR enables retrieval with frozen LLMs but achieves relatively modest effectiveness, while ZeroGR improves generalization through model-generated semantic docids and retrieval-specific adaptation, but still relies on trained components and corpus-specific parameter optimization. These observations lead to the following research question:

What design enables off-the-shelf LLMs to serve as effective generative retrievers without retriever-specific training?

Directly removing the training step. A straightforward attempt at training-free GR adopts a standard GR pipeline, removes the training stage, and uses a pretrained LLM to generate docids under a trie constraint over the corpus docid set. Unfortunately, as shown in Section 2.2, simply removing the training stage does not yield a practical training-free system. The core challenge is a fundamental mismatch: IR demands outputs with valid docids, whereas LLMs are designed for free-form generation. Consequently, models effective in unconstrained settings often fail under constrained decoding, producing an invalid first token.

Aligning data to fit the model’s generative behavior. To bridge this gap, we propose a paradigm shift toward *data-fits-model*: instead of fine-tuning models to fit data, retrieval data and interfaces are restructured to align with the intrinsic generation behavior of strong LLMs without updating model parameters.

We introduce FreeGR, a plug-and-play framework for any off-the-shelf LLM (Figure 1(b)). Unlike training-based GR approaches that rely on task-specific fine-tuning to compensate for weak pretrained priors, FreeGR is designed to directly exploit the strong knowledge and generation capabilities already present in modern LLMs. This allows FreeGR to immediately benefit from stronger foundation models without retriever-specific retraining. FreeGR systematically addresses three components of GR: (i) *Output docids*: the model self-annotates the corpus, generating textual docids that match its natural generation patterns; (ii) *Input context*: prompts and in-context examples are designed to match the model’s free-generation style, prepending preferred preambles to guide it into a “docid list” mode; and (iii) *Decoding strategy*: FreeGR provides a quality-oriented depen-

dency decoding mode and an efficiency-oriented constrained beam search mode, enabling a controllable quality-latency trade-off.

By leveraging the inherent representational and reasoning capabilities of pretrained LLMs, FreeGR can achieve retrieval effectiveness competitive with training-based generative retrievers while exposing a controllable quality-latency trade-off at decoding time. With dependency decoding as the default mode, FreeGR prioritizes coherent listwise generation and top-rank precision. With constrained beam search, FreeGR provides a more efficient alternative for latency-sensitive or large- K retrieval. Importantly, switching between the two modes requires no retriever-specific retraining, corpus re-annotation, or modification to the underlying LLM.

Beyond performance gains, FreeGR removes the need for retriever-specific parameter optimization and enables rapid deployment with different LLM backbones. It is particularly attractive when retriever training resources are limited and query-document relevance can be inferred from semantic cues available to the pretrained LLM.

Experiments. Across multiple BEIR out-of-domain retrieval tasks (Thakur et al., 2021), FreeGR achieves competitive performance against strong embedding and generative retrieval methods without retriever-specific parameter optimization. Furthermore, on NQ (Kwiatkowski et al., 2019) and MS MARCO (Bajaj et al., 2016), FreeGR remains competitive with several trained dense and generative retrievers while requiring no retriever-specific parameter updates.

2 Background and exploration

GR aims to retrieve a target document by generating its identifier given a query (Tay et al., 2022; Bevilacqua et al., 2022; Li et al., 2023). For a corpus $\mathcal{D} = \{d_1, \dots, d_N\}$ with N documents, GR assigns each document $d_i \in \mathcal{D}$ a unique docid $z_{d_i} = \text{id}(d_i)$, represented as a token sequence $\{z_{d_i}^1, \dots, z_{d_i}^T\}$ with maximum length T . The full docid set is $\mathcal{Z} = \{z_{d_1}, \dots, z_{d_N}\}$. We denote the supervised retrieval set as $\mathcal{D}_{\text{retrieval}} = \{(q_i, d_i^+)\}_{i=1}^M$, where d_i^+ is the labeled relevant document for query q_i . A generative LLM $\mathcal{M}$ then models the conditional distribution over docids given the query q , i.e., $p_{\mathcal{M}}(z \mid q)$.

2.1 Generative retrieval with training

To approximate the true relevance distribution, existing GR methods typically fine-tune the pretrained LLM using a combination of indexing

and retrieval training tasks and losses, i.e., $\mathcal{L}_s = \mathcal{L}_{\text{index}} + \mathcal{L}_{\text{retrieval}}$ (Tay et al., 2022).

Indexing. Given a document d_i , the model learns to generate its corresponding identifier z_i to memorize the corpus, defined as

$$\mathcal{L}_{\text{index}} = - \sum_{d_i \in \mathcal{D}} \log p_{\mathcal{M}}(z_{d_i} | d_i). \quad (1)$$

Retrieval. The model learns to generate the identifiers of relevant documents given a query to capture query-document relevance, defined as

$$\mathcal{L}_{\text{retrieval}} = - \sum_{(q_i, d_i^+) \in \mathcal{D}_{\text{retrieval}}} \log p_{\mathcal{M}}(z_{d_i^+} | q_i). \quad (2)$$

In practice, standard GR methods often require ten or more training epochs on large-scale datasets to fit the empirical docid distribution. However, this often causes the model to overfit the training distribution and generalize poorly, leading to sharp performance drops on out-of-domain (OOD) queries and documents. As shown in Figure 5 (in Appendix D), on the NQ dataset, the performance of three heavily post-hoc trained GR baselines (DSI, Tay et al. 2022; SEAL, Bevilacqua et al. 2022; MINDER, Li et al. 2023) degrades sharply under distribution shifts. Compared to in-domain queries, their Hits@5 score on OOD queries drops 55.9% on average. For unseen documents, DSI (which uses numeric docids) achieves near-zero Hits@5, while SEAL and MINDER (which uses textual docids) retain some ability but still suffer a 39.0% average drop versus seen documents.

Recent studies suggest that LLMs have already acquired substantial general knowledge during pre-training. To improve generalization while reducing the training burden, ZeroGR (Sun et al., 2025) uses model-generated semantic docids and synthetic retrieval data together with retrieval-specific adaptation. Unlike standard GR, it avoids conventional indexing training over corpus-docid pairs, but still relies on trained generation components and corpus-specific parameter optimization. Thus, ZeroGR substantially reduces the training burden of GR, while remaining different from the fully frozen retriever considered in this work.

2.2 Generative retrieval without training

We first explore whether a pretrained LLM can directly serve as a generative retriever without retriever-specific parameter optimization. As the simplest attempt, we remove all retrieval-specific

Model	BEIR-NQ		BEIR-FEVER	
	EM	SM	EM	SM
Llama3.1-8B (Free)	2.7	47.2	8.1	75.6
Llama3.1-8B (Cons.)	0.0	—	0.1	—
Qwen3-14B (Free)	6.2	52.8	2.3	78.5
Qwen3-14B (Cons.)	0.3	—	0.1	—

Table 1: Exact Match (EM) and Semantic Match (SM) Rates (%) under NQ and FEVER datasets from BEIR in a preliminary training-free GR attempt.

parameter updates from a standard GR framework (Tay et al., 2022). We evaluate this method on the NQ and FEVER datasets from the BEIR benchmark, whose corpora are derived from Wikipedia, a domain where LLMs exhibit strong prior knowledge. Retrieval is conducted by instantiating the pretrained LLM directly as the generative retriever, using Wikipedia page titles as docids.¹ We compare *constrained decoding* over the docid space with a *free decoding* variant. Performance is measured using Exact Match (EM), and for free decoding we additionally report Semantic Match (SM) to account for non-exact but semantically correct generations.

Free vs. constrained decoding. As shown in Table 1, under free decoding, the model achieves high SM, indicating that a well-pretrained LLM already possesses substantial retrieval-relevant semantic knowledge without task-specific fine-tuning. However, this semantic capability does not translate into EM, as free-form outputs often deviate from valid docids. Constrained decoding enforces valid docid generation and should, in principle, resolve inexact matching. Yet, it leads to unexpectedly poor EM performance.

This occurs because forcing the LLM to produce a valid title-initial token at the first decoding step misaligns with its habitual generation patterns. Figure 2 shows that the model still tends

to follow query-agnostic leading patterns, which allows a few high-probability tokens to dominate first-step generation (see Appendix E for details). Therefore, constrained decoding can prevent the

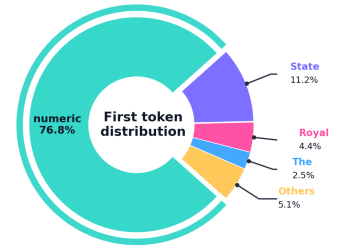


Figure 2: Distribution of the first generated token for different queries on the NQ dataset.

¹This setup is conceptually related to Few-Shot GR, which reconstructs textual docids. In contrast, we directly use document titles as identifiers.

model’s inherent retrieval capability from being effectively expressed. We attribute this failure to a *mismatch between the LLM’s habitual generation patterns and the restricted docid space*.

3 Method

As discussed above, training-free GR should account for the LLM’s natural generation behavior. We revisit the retrieval data and interaction interface to better align them with the model’s pre-trained priors, thereby unlocking its inherent retrieval capability. Crucially, FreeGR requires neither architectural changes nor retriever-specific parameter optimization. It aligns three aspects of the retrieval interface with the frozen LLM: (i) the docids defining the output space, (ii) the input context, and (iii) the decoding strategy. We summarize our method in Algorithm 1 in Appendix I.

3.1 Docid generation

What docids do LLMs naturally prefer to generate? Numerical docids, such as integers or multi-digit strings, are ill-suited for LLMs because they are difficult to generate and interpret without training. In contrast, semantic word-based docids like document titles align naturally with LLMs’ language modeling strengths. However, titles only align with the model’s output habits in corpora it is highly familiar with (e.g., Wikipedia or books) (Bevilacqua et al., 2022; Li et al., 2023). In many specialized domains, titles have complex structures unfamiliar to the model, and some corpora lack titles altogether. The simple solution is to let the LLM $\mathcal{M}$ itself annotate the corpus with docids.

LLM self-annotated docids. We first sample a subset $\mathcal{D}_{\text{sub}} \subset \mathcal{D}_{\text{retrieval}}$ to obtain a set of example LLM-generated docids. To make these docids easier to be hit at retrieval time, we directly use the retrieval prompt P_r (see Appendix A) when generating them. We perform this step in a free-decoding setting, where P_r instructs $\mathcal{M}$ to produce a short, phrase-like docid $z_{d_i}^{\text{sub}}$ conditioned on each query q_i in $\mathcal{D}_{\text{sub}}$ and its labeled relevant document d_i^+ :

$$z_{d_i}^{\text{sub}} = \mathcal{M}(P_r, (q_i, d_i^+)), \quad (q_i, d_i^+) \in \mathcal{D}_{\text{sub}}. \quad (3)$$

Next, for all documents in $\mathcal{D}$, we use several $(z_{d_i}^{\text{sub}}, d_i)$ pairs as in-context examples E_a to prompt $\mathcal{M}$ to annotate every document $d_i \in \mathcal{D}$ as

$$z_{d_i} = \mathcal{M}(P_r, E_a, d_i), \quad d_i \in \mathcal{D}. \quad (4)$$

In full-corpus annotation (Eq. 4), we do not use query-document pairs as in Eq. 3. Instead, we rely

only on the documents themselves, since in realistic scenarios not every document is associated with a relevant query. This ensures that the entire identifier space is defined by strings the LLM naturally tends to generate.

3.2 Prompt construction

What prompts can steer LLMs into retrieval mode? LLMs do not inherently recognize that they are performing a retrieval task; they rely on the prompt to infer the desired behavior (Reynolds and McDonell, 2021; Brown et al., 2020). However, even with strong prompt instructions, LLMs often begin with a short preamble (e.g., “Here are some relevant documents: 1.”) before generating the docid list. Therefore, effective prompts should use the model’s preferred leading pattern, guiding it seamlessly into docid list generation at the point where constraints are enforced.

Input prompts aligned with the model’s free-generation habits. We randomly sample 20 queries from $\mathcal{D}_{\text{sub}}$ and inspect how $\mathcal{M}$ responds to the retrieval prompt P_r under free decoding, obtaining outputs $y(q_i)$. We then ask GPT-4o to analyze these outputs and extract a frequently occurring pattern p that consistently appears right before the docid list, which we refer to as the *preamble pattern* (see Appendix B).

Next, we concatenate each $(q_i, d_i) \in \mathcal{D}_{\text{sub}}$ with its corresponding docid z_{d_i} and the preamble pattern p to form $(q_i \parallel p \parallel z_{d_i})$, which serves as the in-context examples E_r for retrieval. The final input is constructed as

$$(P_r \parallel E_r \parallel q_r \parallel p), \quad (5)$$

where q_r denotes the real query. This design guides the model into “docid list” generation mode.

3.3 Decoding strategies

What decoding strategy should a training-free LLM retriever use? Traditional GR methods often perform constrained beam search over the identifier space to generate multiple docids in parallel, treating beam scores as reliable ranking signals (Tay et al., 2022; Bevilacqua et al., 2022; Li et al., 2023). However, pretrained LLMs are not optimized with ranking-oriented objectives, and thus beam scores may not provide stable relevance rankings in a training-free setting. At the same time, LLMs are naturally good at generating coherent lists in natural language, where earlier items influence the generation of subsequent ones. Accordingly, FreeGR supports two decoding modes

for different deployment scenarios: (i) a quality-oriented docid dependency decoding mode, as the default decoding strategy, since it better follows the coherent list-generation behavior of pretrained LLMs; and (ii) an efficiency-oriented constrained beam search mode, alternative for latency-sensitive or large- K retrieval.

Quality-oriented dependency decoding. FreeGR generates top- K docids sequentially by $\mathcal{M}$:

$$(z_1 \parallel c \parallel z_2 \parallel c \parallel \dots \parallel z_K), \quad (6)$$

where z_j denotes the docid of the j -th retrieved document, and c is a predefined natural-language connector that links consecutive docids (see Appendix B). Constraints $T = \text{Trie}(\mathcal{Z})$ are applied only to the spans corresponding to docids. Before generating each z_j , the trie pointer is reset to the root, and constrained decoding proceeds along the trie. Once z_j is fully generated, its branch is marked as used and excluded from further expansions. The connector c is then appended after z_j , and decoding continues to generate z_{j+1} . In this mode, ranking is determined by the order of docids in a coherent generated list rather than by beam-search scores over a flattened token space.

Efficiency-oriented beam search. For latency-sensitive or large- K retrieval scenarios, FreeGR can instead perform constrained beam search over the same LLM-aligned docid space. Given the same input prompt, this mode applies trie-based constraints during decoding and generates multiple candidate docids in parallel through beam expansion. The final ranked list is obtained according to beam scores:

$$(z_1, z_2, \dots, z_K) = \text{BeamSearch}(\mathcal{M}, x, T, K), \quad (7)$$

where x denotes the constructed retrieval input and $T = \text{Trie}(\mathcal{Z})$ denotes the constrained docid space. Compared with dependency decoding, constrained beam search may be less aligned with the LLM’s natural generation preference, but it substantially reduces inference latency by generating candidates in parallel. Importantly, both decoding modes use the same self-annotated docid space, prompt construction, and frozen LLM backbone. Switching between dependency decoding and constrained beam search requires no retraining or re-annotation.

4 Experimental Setup

Datasets. We evaluate FreeGR in two settings: out-of-domain retrieval and in-domain retrieval, both

using only a small number of labeled target-domain pairs for fixed preprocessing without parameter optimization. For the out-of-domain setting, we use the BEIR benchmark (Thakur et al., 2021), which covers a wide range of retrieval tasks and domains, and which is commonly used to assess cross-domain generalization. Because most dense retrievers (Wang et al., 2022a; Chen et al., 2024) are likely to have been trained on Wikipedia to some extent, we exclude tasks that closely resemble Wikipedia page retrieval (NQ, HotpotQA, DBPedia) in the following experiments to avoid unfair comparisons caused by potential data leakage, following (Sun et al., 2025). For each target domain, FreeGR uses 50 labeled query–document pairs to construct the fixed support set $\mathcal{D}_{\text{sub}}$ used during preprocessing. When an official training split is available, these pairs are sampled from that split; otherwise, we construct a small support subset from the released labeled queries and evaluate exclusively on the remaining disjoint queries. For the in-domain setting, we use two standard retrieval datasets: (i) Natural Questions (NQ) (Kwiatkowski et al., 2019), an open-domain QA retrieval dataset constructed from real search logs where documents are drawn from Wikipedia. (ii) MS MARCO (Bajaj et al., 2016), a large-scale retrieval dataset built from Bing search logs. We adopt the passage retrieval setting.

Baselines and metrics. In the *out-of-domain retrieval* setting, we compare FreeGR against the following representative baselines: (i) Traditional sparse retrieval: BM25 (Robertson and Zaragoza, 2009), which requires no training and naturally supports out-of-domain retrieval as a classic baseline; (ii) Single-task dense retrievers trained on MS MARCO: Contriever-MARCO (Izacard et al., 2021) and GTR (Ni et al., 2022), all trained on MS MARCO; (iii) General-purpose dense retrievers: E5 (Wang et al., 2022a), BGE (Chen et al., 2024), OpenAI-Embedding (OpenAI, 2024), E5-Mistral (Wang et al., 2024), and GritLM (Muenighoff et al., 2024), which achieve strong cross-domain generalization via contrastive learning on large-scale multi-domain corpora; (iv) Generalizable GR: Few-Shot GR (Askari et al., 2024), and ZeroGR (Sun et al., 2025).

In the *in-domain retrieval* setting, we compare FreeGR against data-driven retrieval methods that use in-distribution supervised training, including: (i) Dense retrieval with pretrained embedding backbones (DPR-style (Karpukhin et al.,

Method	Avg.	Fever	SciFact	SciDocs	Covid	ArguAna	NFCorpus	FiQA	Touche	TREC-News	Quora	CQADupStack
BM25	42.3	58.3	65.1	12.4	58.3	32.7	32.7	24.8	59.0	20.7	73.8	28.0
Contriever	47.6	90.6	70.3	15.7	59.6	32.1	33.9	35.5	42.5	27.3	86.6	29.9
GTR-T5-base	45.3	83.2	58.6	12.1	56.1	32.7	32.7	34.5	48.1	22.5	88.7	28.9
GTR-T5-large	48.0	86.8	61.9	12.8	56.7	34.3	33.4	43.3	53.1	26.6	89.1	29.8
E5-Base	48.9	91.1	73.9	18.3	60.7	31.1	35.0	39.6	41.1	22.9	86.4	38.3
E5-Large	49.2	93.1	76.3	19.6	55.2	31.7	37.4	42.3	34.8	25.3	86.9	38.5
BGE-Base	50.5	85.6	74.3	19.8	67.1	41.8	36.0	43.4	41.4	21.2	89.8	35.1
BGE-Large	51.8	86.6	75.2	20.6	64.5	41.6	38.1	48.5	45.5	21.4	89.3	38.3
E5-mistral-7B	55.7	91.8	76.6	20.7	77.8	44.1	39.6	59.7	46.8	29.4	84.8	41.4
GritLM-7B	45.0	68.9	76.8	19.6	31.5	40.7	36.9	44.1	21.5	34.9	84.9	35.0
OpenAI Embed	54.2	92.8	73.1	21.2	78.2	37.1	37.6	48.5	47.5	26.2	89.9	44.1
Few-Shot GR (Llama3-8B)	8.3	18.3	11.0	3.3	12.5	5.8	5.7	9.1	1.6	3.8	15.4	4.9
ZeroGR (Llama-3B)	48.1	86.7	72.8	18.7	73.5	35.4	34.7	34.1	37.5	23.5	76.7	35.2
FreeGR (Qwen3-4B)	36.6	86.4	69.9	10.2	43.7	33.5	17.5	28.1	27.0	18.6	47.3	20.4
FreeGR (Llama2-13B)	47.4	89.3	72.8	19.5	74.1	33.0	32.8	37.6	34.1	18.9	75.5	34.0
FreeGR (Qwen3-14B)	48.5	92.0	74.1	20.2	71.1	35.1	36.5	37.2	32.6	24.7	75.9	33.7

Table 2: Out-of-domain retrieval performance (nDCG@10) on the BEIR benchmark. FreeGR achieves competitive cross-domain performance without any parameter updates.

2020)): we train dense retrievers via contrastive learning with BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) backbones. We do not use more advanced retrieval-specific embedding models to avoid data leakage. (ii) Generative retrieval: DSI (Tay et al., 2022), DSI-QG (Zhuang et al., 2022), SEAL (Bevilacqua et al., 2022), LTRGR (Li et al., 2024), MINDER (Li et al., 2023), TSGen (Zhang et al., 2024), RIPOR (Zeng et al., 2024a), PAG (Zeng et al., 2024b), and Few-Shot GR (Askari et al., 2024).

We conduct FreeGR with two well-trained LLM backbones: Qwen3 (Yang et al., 2025) and Llama2 (Touvron et al., 2023). Following existing GR works (Tay et al., 2022; Bevilacqua et al., 2022), we use Hits@ k (H@ k), nDCG@ k and MRR@ k as evaluation metrics. Details of the implementation are provided in Appendix C.

5 Experimental Results

We evaluate FreeGR from multiple perspectives, including out-of-domain and in-domain retrieval performance, design ablations, decoding strategies, and scaling behavior with respect to model size and corpus size.

5.1 Out-of-domain retrieval results

We evaluate FreeGR on the BEIR benchmark without retriever-specific parameter optimization, comparing it against strong generalization-oriented retrieval methods. As shown in Table 2, FreeGR achieves competitive overall performance. The 4B model performs strongly on several datasets (e.g., FEVER, SciFact, and ArguAna) but is weaker on others, reflecting its dependence on backbone

capacity. With larger backbones (13B and 14B), FreeGR reaches performance comparable to ZeroGR and several strong dense retrievers trained on large-scale corpora. Importantly, scaling FreeGR to stronger backbones requires no retriever-specific parameter optimization, allowing it to directly benefit from advances in foundation models. To disentangle the effect of backbone capacity from that of the retrieval method, we additionally evaluate FreeGR using the same Llama-3B backbone as ZeroGR. Under this controlled setting, FreeGR achieves an average BEIR nDCG@10 of 34.5, substantially outperforming Few-Shot GR (6.7), while remaining below ZeroGR (48.1). This result shows that aligning the docid space, input context, and decoding interface substantially improves the retrieval effectiveness of a frozen LLM, while corpus-specific parameter optimization can still provide additional gains, particularly for smaller backbones.

Furthermore, we categorize the tasks in Table 2 according to the degree of familiarity of the LLM with the underlying corpus: (i) *Full prior*, where the corpus consists of Wikipedia pages (FEVER); (ii) *Partial prior*, where the corpus is composed of books or academic papers (SciFact, SciDocs, and TREC-COVID); and (iii) *Low prior*, where the corpus is derived from Web data (the remaining tasks). As shown in Figure 4, FreeGR performs best in domains where the LLM possesses strong prior knowledge. In the *Full prior* group, FreeGR achieves higher performance than ZeroGR, while the gap narrows as the amount of prior knowledge decreases. This trend reflects FreeGR’s dependence on the semantic knowledge available in the frozen backbone. Nevertheless, even in the

Method	NQ		MS MARCO	
	H@5	H@20	H@10	MRR@10
BM25	43.6	62.9	45.1	18.5
BERT-Large	66.3	78.1	64.5	31.4
RoBERTa-Large	65.2	80.6	68.7	34.3
DSI	54.3	66.3	43.6	23.4
DSI-QG	35.5	52.7	—	10.5
SEAL	63.5	79.3	61.1	37.2
LTRGR	68.8	80.3	—	25.5
MINDER	66.0	80.0	64.7	37.9
TSGen	67.1	79.7	64.0	35.1
RIPOR	—	—	56.2	33.3
PAG	—	—	67.0	38.5
Few-Shot GR	12.8	15.0	6.1	3.5
FreeGR(Qwen3-4B)	60.9	71.5	57.3	31.6
FreeGR(Llama2-13B)	65.2	73.6	61.9	34.5
FreeGR(Qwen3-14B)	64.9	75.3	62.5	35.2

Table 3: In-domain retrieval results on the NQ and MS MARCO datasets. FreeGR remains competitive while requiring no in-domain post-training.

Low prior group, FreeGR remains competitive with ZeroGR without retriever-specific parameter optimization. These results also reveal an important applicability boundary: FreeGR is better suited to domains where query–document relevance can be inferred from semantic cues available to the pre-trained LLM.

5.2 In-domain retrieval results

We further evaluate FreeGR on two popular retrieval datasets with training splits, NQ and MS MARCO. We mainly compare FreeGR against dense and generative retrieval methods that use in-domain supervised training. Note that SOTA dense embedding retrievers are often trained on NQ and MS MARCO. To avoid data leakage we re-train dense retrievers from scratch using BERT and RoBERTa backbones. The results are summarized in Table 3. FreeGR achieves strong retrieval performance on both NQ and MS MARCO. For example, on Hits@10, FreeGR outperforms trained GR methods such as DSI and SEAL. While some trained methods still lead on certain metrics, FreeGR provides a competitive alternative with a fully frozen backbone and requires no retriever-specific parameter optimization.

5.3 Breakdown analysis

Docid generation. We consider four types of docid designs: (i) document titles; (ii) docids annotated by a local LLM (Qwen-14B); (iii) docids annotated by an external stronger LLM (GPT-4o); and (iv) FreeGR (self-annotated docids). We evaluate them on three datasets whose document metadata include titles: viz. FEVER, SciFact, and TREC-News. The

Docid source	Fever	SciFact	TREC-News
Title	92.5	25.7	3.0
Annotated by Llama3	91.4	70.5	23.6
Annotated by GPT-4o	92.1	74.5	25.2
FreeGR	92.0	74.1	24.7

Table 4: Effect of different docid designs on retrieval performance (nDCG@10).

Setting	Preamble p	ICL E_r	BEIR Avg
w/o p , w/o E_r	✗	✗	0.1
E_r only	✗	✓	0.1
p only	✓	✗	46.3
$p+E_r$ (FreeGR)	✓	✓	48.5

Table 5: Ablation on input construction for FreeGR on BEIR (nDCG@10).

title styles of these three datasets differ substantially: FEVER titles come from Wikipedia pages and are typically entity-centric; SciFact titles are highly scientific; while TREC-News titles are usually more colloquial. This diversity allows us to examine the robustness of different constrained docid spaces. As shown in Table 4, raw titles are effective for familiar entity-centric corpora such as FEVER, but degrade substantially when titles are ambiguous or less generation-friendly, such as in SciFact and TREC-News. In contrast, LLM-annotated docids achieve consistently strong performance, and docids generated by either the retriever itself or other local/stronger external models remain competitive. This suggests that FreeGR’s docid design is highly transferable: docids annotated once with a single backbone can often be reused across retrievers, reducing the need for corpus re-annotation and lowering data preparation costs.

Input prompt construction. We ablate two key input components: the preamble pattern p and the in-context examples E_r . We evaluate four input configurations by averaging performance over all BEIR tasks: (i) w/o p , w/o E_r : the input contains only the retrieval prompt P_r and the query; (ii) E_r only: the input contains P_r , the query, and a few examples that include the preamble pattern; (iii) p only: the input contains P_r , the query, and the preamble pattern; and (iv) $p+E_r$ (FreeGR). As shown in Table 5, removing p causes performance to nearly collapse even with E_r , while adding p alone yields strong results, and combining p with E_r performs best. These results indicate that the preamble pattern is the critical factor for resolving the first-token mismatch by switching the model into a docid-generation mode, with in-context examples providing additional refinement once this mode is established.

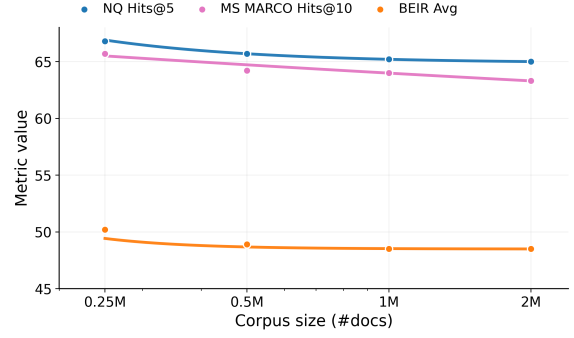
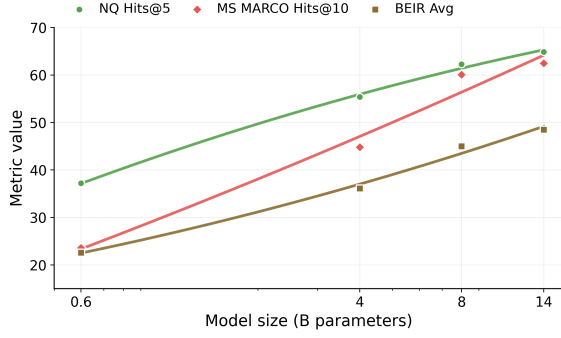


Figure 3: Scaling behavior of FreeGR. (Left) Retrieval improves consistently with larger LLMs. (Right) Performance degrades only slightly as the candidate set grows.

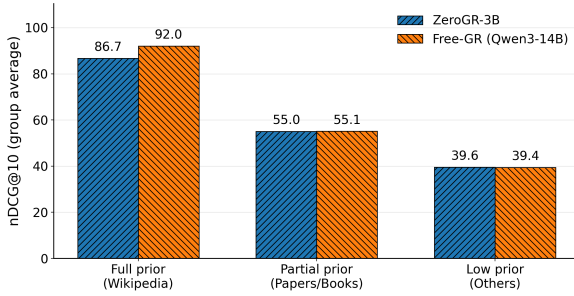


Figure 4: BEIR task-group results by model familiarity.

Decoding-mode trade-off. We compare dependency decoding and constrained beam search on NQ. Table 6 reports Hits@ K and normalized latency for different values of K . Dependency decoding consistently achieves higher Hits@ K , indicating that coherent listwise generation produces more stable ranking signals and better aligns with the natural list-generation behavior of LLMs. In contrast, constrained beam search is substantially more efficient, as parallel candidate expansion leads to slower latency growth with K . Overall, the two decoding modes are complementary: dependency decoding is better suited for quality-focused scenarios with small K (e.g., open-domain QA (Chen et al., 2017) and recommendation (Covington et al., 2016)), while constrained beam search is preferable for latency-sensitive or large- K retrieval (e.g., web search and e-commerce (Asadi and Lin, 2013; Li et al., 2021)). This demonstrates that FreeGR enables a controllable quality–latency trade-off at decoding time.

5.4 Scaling data and model size

Model scaling. We evaluate FreeGR of varying sizes on NQ, MS MARCO, and BEIR. As shown in Figure 3 (Left), performance scales consistently with model capacity. This gain reflects stronger instruction following and generalization in larger LLMs, underscoring the value of model scaling for GR once the mismatch between free generation

K	Hits@ K		Latency	
	Dep.	BS	Dep.	BS
1	49.3	49.3	1 ×	1.04 ×
5	64.9	61.4	6.38 ×	1.29 ×
20	75.3	71.0	24.41 ×	1.51 ×
50	76.0	71.2	76.02 ×	2.47 ×

Table 6: Performance-efficiency trade-off on the NQ dataset under different top- K .

and constrained decoding is resolved.

Data scaling. We scale the candidate document set (and thus the size of the constrained docid space) to assess robustness on larger corpora, using Qwen3-14B across NQ, MS MARCO, and BEIR. When the dataset corpus is smaller than the target size, we truncate to the full corpus. Figure 3 (Right) shows only a minor performance drop as corpus size grows. The stability stems from FreeGR’s semantic, query-driven docid generation rather than representation matching. This trend suggests that, as long as docids in the corpus are sufficiently discriminative, FreeGR can still operate well on very large corpora.

6 Related Work

Sparse retrieval like BM25 (Robertson and Zaragoza, 2009) are training-free and robust under distribution shifts but rely on term overlap. Dense retrieval (Karpukhin et al., 2020; Formal et al., 2021; Xiong et al., 2020) uses learned embeddings to capture semantic similarity, yet its cross-domain performance depends heavily on massive and heterogeneous corpora. Standard GR relies on supervised learning to memorize the corpus, making it training-intensive and sensitive to distribution shifts (Tay et al., 2022; Bevilacqua et al., 2022). ZeroGR (Sun et al., 2025) reduces the training burden through model-generated semantic docids and retrieval-specific adaptation, but still relies on trained components and corpus-specific parameter optimization. FreeGR is closely related to ZeroGR

in that both exploit model-generated semantic docids and constrained generation. However, ZeroGR trains a separate docid generator and a corpus-specific generative retriever, whereas FreeGR uses the same frozen target LLM for corpus annotation and retrieval without retriever-specific parameter optimization. Fully training-free approaches (Askari et al., 2024) attempt to directly use pre-trained LLMs for retrieval, but suffer from a mismatch between the LLM’s habitual generation patterns and the restricted docid space. In contrast, we restructure the data to align data with the model’s intrinsic generation habits. Detailed related work is discussed in Appendix G.

7 Conclusion

This paper revisits generative retrieval from a training-free perspective, aiming to directly leverage off-the-shelf LLMs without retriever-specific parameter optimization. We identify a key challenge: a mismatch between constrained docid generation and the free-form generation behavior of pretrained LLMs. To address this, we propose FreeGR, a data-fits-model interface-alignment framework that aligns docid design, prompting, and decoding with the intrinsic generation behavior of frozen LLMs. FreeGR further supports a controllable effectiveness–efficiency trade-off via alternative decoding strategies. Experiments on BEIR, NQ, and MS MARCO show that FreeGR achieves competitive retrieval performance across diverse settings and scales to different model sizes and corpus sizes without retriever-specific parameter updates.

Limitations

The limitations of this work are as follows: (i) FreeGR relies on the capability of the pretrained LLM. When the backbone model is relatively weak, or when the target domain lacks sufficient prior knowledge in the pretraining data, the retrieval performance may degrade. (ii) FreeGR involves a quality-efficiency trade-off at decoding time. The default dependency decoding mode provides stronger listwise ranking but incurs higher latency for large K , while the efficiency-oriented constrained beam search mode reduces latency but may sacrifice some retrieval effectiveness due to less reliable beam-score ranking. Thus, the decoding mode should be selected according to deployment requirements. (iii) Since constrained decoding is difficult to implement effectively when accessing

stronger online models (e.g., GPT-5) through APIs, our evaluation does not include such models. (iv) While FreeGR mitigates the first-token mismatch problem, we do not explicitly investigate potential decoding mismatches that may arise in later generation steps. Studying such effects remains an interesting direction for future work.

Ethical Considerations

This work analyzes baselines and our newly proposed method using existing benchmark datasets and reports aggregate results only. It does not collect new user data or involve human subjects.

Acknowledgements

This work was funded by the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Natural Science Foundation of China (NSFC) under Grants No. U25B2076, 62472408 and 62441229, and the National Key Research and Development Program of China under Grants No. 2023YFA1011602. This research was also (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union under grant agreement No. 101201510 (UNITE). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

- Nima Asadi and Jimmy Lin. 2013. Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 997–1000.
- Arian Askari, Chuan Meng, Mohammad Aliannejadi, Zhaochun Ren, Evangelos Kanoulas, and Suzan Verberne. 2024. Generative retrieval with few-shot indexing. *arXiv preprint arXiv:2408.02152*.
- 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.

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.
- Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Continual learning for generative retrieval over dynamic corpora. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 306–315.
- Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 191–200.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.
- Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 3(8):4171–4186.
- Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.
- Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.
- 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.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3181–3189.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. *arXiv preprint arXiv:2305.16675*.
- Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to rank in generative retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8716–8723.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.
- Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. DSI++: Updating transformer memory with new documents. *arXiv preprint arXiv:2212.09744*.
- Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and 1 others. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

- OpenAI. 2024. New embedding models and API updates. OpenAI Blog. Jan 25, 2024.
- Ronak Pradeep, Kai Hui, Jai Gupta, Adam D. Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q. Tran. 2023. How does generative retrieval scale to millions of passages? *arXiv preprint arXiv:2305.11841*.
- Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). volume 21, pages 140:1–140:67.
- Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*, pages 1–7.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Found. Trends Inf. Retr.*, 3:333–389.
- Weiwei Sun, Keyi Kong, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, Zhaochun Ren, and Yiming Yang. 2025. ZeroGR: A generalizable and scalable framework for zero-shot generative retrieval. *arXiv preprint arXiv:2510.10419*.
- Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. 2024. List-wise generative retrieval models via a sequential learning process. *ACM Transactions on Information Systems*, 42(5):1–31.
- 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.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.
- Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, and 1 others. 2022b. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.
- Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou. 2023. Novo: Learnable and interpretable document identifiers for model-based IR. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2656–2665.
- Shiguang Wu, Zhaochun Ren, Xin Xin, Jiyuan Yang, Mengqi Zhang, Zhumin Chen, Maarten de Rijke, and Pengjie Ren. 2025. Constrained auto-regressive decoding constrains generative retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2429–2440.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- 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. *arXiv preprint arXiv:2310.12455*.
- Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2024a. Scalable and effective generative information retrieval. In *Proceedings of the ACM Web Conference 2024*, pages 1441–1452.
- Hansi Zeng, Chen Luo, and Hamed Zamani. 2024b. 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*, pages 469–480.
- 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.
- 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 Retrieval Prompt

[Retrieval Prompt Pr]

You are a retrieval assistant.

Your task is to output the possible titles of documents related to the query.

Each document title consists of a few short phrases that accurately capture the core content of the document, where more important phrases should appear earlier. A title must contain only lowercase letters, digits, and spaces.

- The input describes information relevant to the query.
- Output a ranked list of document identifiers for the top K most relevant documents.
- Output format: a numbered docid list.
- Do NOT add explanations, descriptions, or any extra text.

Query: <QUERY>

B Input Template

This section presents the preamble pattern p and natural-language connector c used by FreeGR with Qwen3-14B as the backbone:

[Preamble pattern p]

The following documents may be relevant to the user's query:1."

[Natural-language connector c]

";<j>."

C Implementation Details

C.1 Implementation details of FreeGR

For all datasets, we set $|\mathcal{D}_{\text{sub}}| = 50$. When an official training split is available, the query-document pairs are sampled from that split; otherwise, we construct a small support subset from the released labeled queries and evaluate on the remaining disjoint queries. Sampling is performed without replacement using a fixed random seed. These labeled pairs are used only for fixed preprocessing and in-context demonstrations, without any gradient-based parameter optimization.

In docid self-annotation, we randomly select five examples from $\mathcal{D}_{\text{sub}}$ to construct E_a and use greedy decoding with the default decoding hyperparameters of the corresponding LLM. For each model and task, the preamble pattern p is automatically induced by GPT-4o from the free-decoding outputs of 20 queries sampled from $\mathcal{D}_{\text{sub}}$. During retrieval inference, we use a single fixed ICL example ($|E_r| = 1$), perform greedy decoding, and apply a trie constraint over the docid space for each

docid generation span. We append the fixed preamble pattern p after the new query. All experiments are conducted on two NVIDIA H20 GPUs.

C.2 Implementation details of baselines

For the out-of-domain baselines, we report the numbers exactly as provided in ZeroGR (Sun et al., 2025) to ensure a fair comparison under the same evaluation protocol. For the in-domain baselines, we implement a dense retriever with BERT and RoBERTa backbones, as well as three post-trained GR methods (DSI, SEAL, and MINDER) for the analyses in Section D. Results of the remaining methods are directly taken from the corresponding references (Li et al., 2024; Zeng et al., 2024b).

For the dense retriever, we adopt a standard dual-encoder architecture to encode queries and documents independently, and obtain a single-vector representation via pooling. We train the retriever with contrastive learning using in-batch negatives, with a batch size of 64. At inference time, we build a FAISS index over all document embeddings and retrieve top- K documents by vector similarity search.

For the three GR methods, we use Qwen3-4B as a unified backbone to implement DSI, SEAL, and MINDER. This model size is substantially larger than the original backbones used in these methods (around 0.5B), while still being tractable to train. We follow the docid construction procedures described in the original papers to build the docid space, and train the model with teacher-forced sequence-level cross-entropy. During inference, we perform constrained decoding using a trie for DSI and an FM-index for SEAL and MINDER.

C.3 Implementation details of simplest training-free GR

For the simplest training-free GR baseline used in Section 2.2, we instantiate Qwen3-14B as the generative retriever on the BEIR NQ and FEVER tasks by completely removing the training stage of DSI. We use the original document titles to construct the trie constraints and also treat them as the ground-truth for evaluation. Under the constrained decoding setting, we keep the inference-time Trie constrained decoding of DSI. Under the free decoding setting, we do not impose any constraints at all, and thus it can be viewed as the vanilla Qwen3-14B without any modifications.

The evaluation protocol in this experiment differs from our main experiments: we only evaluate the rank-1 output, since it is difficult to obtain a

complete ranked list under free decoding. Specifically, we report (i) Exact Match (EM): whether the generated string exactly matches the title of any relevant document; and (ii) Semantic Match (SM): whether the generated string is semantically equivalent to the title of any relevant document, judged by GPT-4o.

D GR Generalization with Post-training

In this section, we study how traditional GR methods tend to overfit the training distribution due to heavy post-training, examining both query- and document-level generalization. Without loss of generality, we reproduce three representative post-trained GR methods on NQ using Qwen3-4B as the backbone: DSI, SEAL, and MINDER. For query-level generalization, we use GPT-4o to construct an out-of-distribution (OOD) evaluation set of 1,000 query-document pairs, each covering topics absent from the training set, and compare GR performance on this OOD set with the in-distribution (ID) queries in training set. For document-level generalization, we hold out a subset of unseen documents during training and evaluate the same GR methods, comparing results with those on seen documents.

OOD queries. Figure 5(a) shows a sharp performance drop from ID to OOD across all GR methods, indicating that these models often memorize dataset-specific query-to-docid mappings rather than learning robust retrieval behaviors. The degradation aligns with overfitting to the training query distribution: even when the target document is unchanged, rewording queries from a different distribution can severely impair docid generation. This motivates avoiding the “model-fits-data” paradigm under limited data and instead redesigning the data and interface to better use the pretrained model prior.

Unseen documents. Figure 5(b) shows that methods relying on memorized docids and query-document correspondences often fail on unseen documents, revealing brittleness to corpus evolution (e.g., incremental updates). Thus, in new domains or with newly added documents, traditional GR typically requires costly continual training (Chen et al., 2023; Mehta et al., 2022).

E Analysis of First-step Generation

Figure 6 shows that when randomly sampling 20 different queries, the pairwise similarity between

Model Size	Training (s)	Docid annotation (s)
4B	702	78
14B	2832	140

Table 7: Runtime comparison between training-based GR and FreeGR docid annotation on a corpus of 10,000 MS MARCO documents using 4 A800 GPUs.

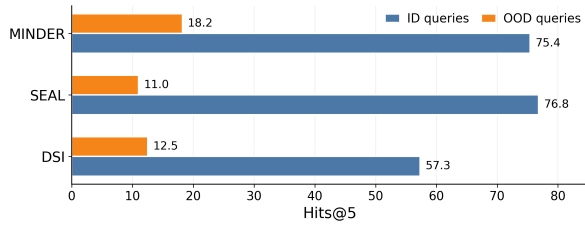
the first-step logit vectors is consistently high (measured by cosine similarity), indicating that the constrained decoder induces an almost query-invariant distribution at the first decoding step. This supports that forcing the LLM to select a valid title-initial token at the first step is misaligned with its natural generation habits and the model still tends to follow query-agnostic leading patterns, causing a few high-probability tokens to dominate the first-step generation and thereby limiting its inherent retrieval capability.

F Cost of docid annotation

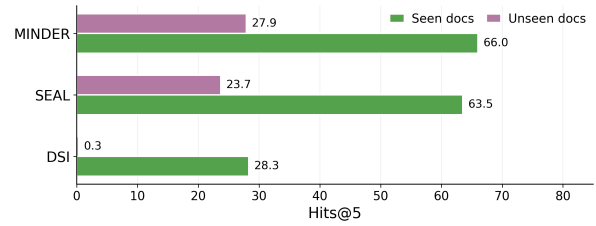
A potential concern of FreeGR is the cost of docid annotation. Although FreeGR eliminates the need for training, it requires using a pretrained LLM to annotate the corpus with textual docids once. We perform docid annotation on a corpus of 10,000 MS MARCO documents using four A800 GPUs. The annotation time is then compared with the training time of training-based GR methods (10 epochs including both indexing and retrieval training). Since both processes scale approximately linearly with respect to corpus size, the measured runtime can be used to estimate the cost at larger scales. Table 7 shows that docid annotation is significantly faster than training. Under a linear scaling approximation, annotating the full MS MARCO corpus (8.8M documents) with a 14B model on four A800 GPUs would take less than two days. In contrast, training-based GR at this scale typically require hundreds of GPU hours for training. This highlights the efficiency advantage of FreeGR. Moreover, as shown in Section 4, docids generated by different LLM annotators exhibit strong transferability. This suggests that smaller or distilled annotator models may be used in the future to further reduce the annotation cost.

G Extended Related Work

Sparse retrieval. Traditional sparse retrieval methods, such as BM25 (Robertson and Zaragoza, 2009), score documents through lexical matching and term-level statistics. They require no task-specific training, are inexpensive to deploy, and



(a) Query-level generalization: performance on out-of-distribution (OOD) queries versus in-distribution (ID) queries



(b) Document-level generalization: performance on unseen (held-out) documents versus seen documents.

Figure 5: Generalization of traditional GR on NQ. Post-trained GR methods show substantial drops under distribution shifts.

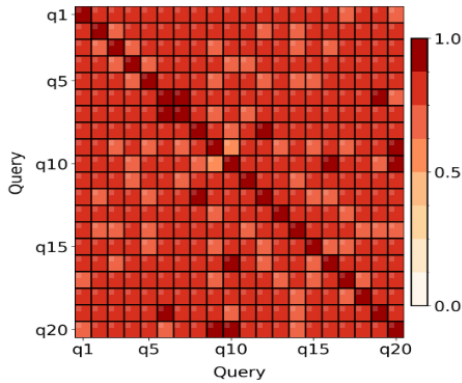


Figure 6: Heatmap of pairwise cosine similarities between the first token logit vectors restricted to the allowed set for 20 sampled queries. The similarities are consistently high.

often remain robust under distribution shifts because they do not learn dataset-specific embedding spaces. However, their reliance on term-level overlap makes them brittle for synonymy, paraphrases, and compositional or multi-facet information needs, where relevant documents may not share exact terms with the query. Some work improves sparse retrieval via learned term reweighting and document expansion (Mao et al., 2020), but the core limitation persists: sparse signals often fail to adequately capture semantics, especially when queries are complex or vocabulary mismatch is severe.

Dense retrieval and cross-domain generalization.

Dense retrieval learns embedding functions that map queries and documents into a shared vector space, enabling semantic matching beyond literal overlap (Karpukhin et al., 2020; Formal et al., 2021; Khattab and Zaharia, 2020). In-domain, dense retrievers can substantially outperform sparse baselines, yet their cross-domain effectiveness is highly sensitive to training data coverage, domain-specific terminology, and annotation quality. To improve out-of-distribution generalization, prior work ex-

plores larger and more heterogeneous pretraining corpora as well as domain-adaptive pretraining (Wang et al., 2022a; Chen et al., 2024). However, dense retrieval typically relies on substantial supervised signals, and its performance can still degrade when the target domain deviates from the training distribution or when the retrieval objective mismatches downstream usage.

Generative retrieval (GR). Generative retrieval reframes retrieval as conditional generation that given a query, the model generates the docids of relevant documents and then maps the generated docids back to the corpus. Key components of GR include docid construction, constrained decoding, and training.

Regarding docids, numerical docids are compact but poorly aligned with an LLM’s output prior, thus using numerical docids typically requires strong post-hoc training (Tay et al., 2022; Wang et al., 2022b). Textual docids (e.g., document titles) are more natural to generate, yet not all corpora provide usable titles (Bevilacqua et al., 2022; Chen et al., 2022). Recent studies have shown that docid design can have a substantial impact on GR performance (Li et al., 2023), suggesting that what the model is asked to generate can be as important as the model itself.

Because retrieval requires exact matching of identifiers, constrained decoding is widely used to restrict generation to valid docids, commonly implemented via a trie (Tay et al., 2022) or FM-index (Bevilacqua et al., 2022). Moreover, to reduce latency, existing GR methods often use beam search to produce multiple candidate docids in parallel (Tay et al., 2022; Bevilacqua et al., 2022; Li et al., 2024). This implicitly assumes that beam scores provide a stable ranking signal, which is not fully consistent with an LLM’s pretraining objective. Some GR work mitigates this issue by intro-

Dataset	#Queries	#Docs
FEVER	6,666	5.42M
SciFact	300	5K
SCIDOCS	1,000	25K
TREC-COVID	50	171K
ArguAna	1,406	8.67K
NFCorpus	323	3.6K
FiQA-2018	648	57K
Touché-2020	49	382K
TREC-NEWS	57	595K
Quora	10,000	523K
CQADupStack	13,145	457K

Table 8: Dataset statistics for the BEIR dataset.

ducing learn-to-rank losses (Li et al., 2024; Tang et al., 2024), however, recent results indicate that constrained beam search remains a key bottleneck for GR (Wu et al., 2025).

Standard GR typically requires substantial training: the model must learn corpus-to-docid and query-to-docid mappings via sequence-level supervision. This makes GR sensitive to distribution shifts (e.g., changes in query style or domain drift) and difficult to scale to new corpora, often requiring carefully designed continual training (Mehta et al., 2022; Chen et al., 2023). Very recently, ZeroGR (Sun et al., 2025) reduces the training burden of GR through model-generated semantic docids, synthetic retrieval data, and retrieval-specific adaptation. Unlike standard GR, it avoids conventional indexing training over corpus–docid pairs, but still relies on trained generation components and corpus-specific parameter optimization. FreeGR is closely related to ZeroGR in that both exploit model-generated semantic docids and constrained generation. However, ZeroGR trains dedicated components for docid generation and retrieval adaptation, whereas FreeGR uses the same frozen target LLM for corpus annotation and retrieval without retriever-specific parameter optimization. Moreover, the two methods differ in listwise decoding: FreeGR retains previously generated docids in the autoregressive context when generating subsequent results, explicitly modeling dependencies among retrieved docids.

H Dataset Statistics

We show the statistics of the BEIR benchmark in Table 8, and the statistics of the NQ and MS MARCO dataset in Table 9.

Dataset	#Docs	#Train pairs	#Test queries
MS MARCO	8.8M	532K	7K
NQ	2.7M	312K	8K

Table 9: Statistics of the NQ and MS MARCO datasets.

I Algorithm of FreeGR

We summarize FreeGR in Algorithm 1.

Algorithm 1 FreeGR

Require: Corpus $\mathcal{D}$, retrieval data $\mathcal{D}_{\text{retrieval}}$, pre-trained LLM $\mathcal{M}$, retrieval prompt P_r , connector c , decoding mode $m \in \{\text{DEP}, \text{BEAM}\}$.

Ensure: Top- K retrieved documents R for an input query q .

Docid generation

- 1: Sample $\mathcal{D}_{\text{sub}} \subset \mathcal{D}_{\text{retrieval}}$
- 2: **for** $(q_i, d_i^+) \in \mathcal{D}_{\text{sub}}$ **do**
- 3: $z_{d_i^+}^{\text{sub}} \leftarrow \mathcal{M}(z \mid P_r, q_i, d_i^+)$
- 4: **end for**
- 5: $E_a \leftarrow \{(d_i^+, z_{d_i^+}^{\text{sub}}) : (q_i, d_i^+) \in \mathcal{D}_{\text{sub}}\}$
- 6: **for** $d_i \in \mathcal{D}$ **do**
- 7: $z_{d_i} \leftarrow \mathcal{M}(P_r \parallel E_a \parallel d_i)$
- 8: **end for**
- 9: $\mathcal{Z} \leftarrow \{z_{d_i} : d_i \in \mathcal{D}\}$
- 10: $T \leftarrow \text{Trie}(\mathcal{Z})$

Prompt construction

- 11: Collect free-decoding outputs $y(q_i)$ of $\mathcal{M}$ on (P_r, q_i) , for $(q_i, d_i^+) \in \mathcal{D}_{\text{sub}}$.
- 12: $p \leftarrow \text{Preamble}(y)$
- 13: $E_r \leftarrow \{q_i \parallel p \parallel z_{d_i^+}^{\text{sub}} : (q_i, d_i^+) \in \mathcal{D}_{\text{sub}}\}$

Decoding

- 14: $x \leftarrow (P_r \parallel E_r \parallel q \parallel p)$
 - 15: **if** $m = \text{DEP}$ **then**
 - 16: $s \leftarrow []$
 - 17: **for** $j = 1$ to K **do**
 - 18: $z_j \leftarrow \mathcal{M}(x, s, T)$
 - 19: $s \leftarrow s \parallel z_j \parallel c$
 - 20: **end for**
 - 21: $R \leftarrow \text{Normalize}(s)$
 - 22: **else**
 - 23: $(z_1, z_2, \dots, z_K) \leftarrow \text{BeamSearch}(\mathcal{M}, x, T, K)$
 - 24: $R \leftarrow \text{Normalize}(z_1, z_2, \dots, z_K)$
 - 25: **end if**
 - 26: **return** R
-