Generative Store Retrieval in Taobao Search

Yingchen Zhang[1,2], Ruqing Zhang[1,2(✉)], Jiafeng Guo[1,2(✉)], Maarten de Rijke[3], Kaixuan Zhang[4], Zhihong Chen[4], Fuyu Lv[4], and Xueqi Cheng[1,2]

[1] State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{zhangyingchen23s,zhangruqing,guojiafeng,cxq}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] University of Amsterdam, Amsterdam, The Netherlands
m.derijke@uva.nl
[4] Hangzhou, China
zhangkaixuan.zkx@taobao.com, {jhon.czh,fuyu.lfy}@alibaba-inc.com

Abstract. With the arrival of large language models, generative retrieval (GR) has emerged as a new paradigm, which generates identifiers of documents relevant to a given query. To deploy this paradigm for store search of Taobao, we address two key challenges: (i) stores are associated with complex store-level metadata and product-related information, which increases the need for high-quality labeled data; and (ii) user queries may either target a specific store or describe a product category to retrieve a diverse set of relevant stores, resulting in varying demands for retrieval precision and diversity. We propose GenStore, a GR framework tailored for e-commerce store search. For training, we synthesize high-quality pseudo-queries from store metadata and affiliated product details, pairing them with store identifiers to learn query-store mappings. During inference, we first classify each query's intent, then apply entropy-gated contrastive decoding that performs constrained generation of store names by contrasting an expert model with a lightweight amateur model. An intent-specific similarity penalty further promotes diversity when appropriate. Extensive offline experiments and online A/B testing demonstrate that GenStore significantly enhances retrieval relevance compared to existing methods while preserving result diversity.

Keywords: Store retrieval · Generative retrieval · E-commerce search

1 Introduction

E-commerce platforms are now integral to daily life. On Taobao, store search serves as a key entry point for users seeking official flagship stores or trusted third-party sellers, directly affecting browsing efficiency and user experience.

K. Zhang, Z. Chen and F. Lv—Independent researcher.

H. Jung et al. (Eds.): DASFAA 2026, LNCS 16540, pp. 606–618, 2026.
https://doi.org/10.1007/978-981-92-0378-9_39

Generative Retrieval. Traditional e-commerce retrieval methods, including sparse [16] and dense retrieval [3,4], are limited in their ability to incorporate general world knowledge and often fail to capture the fine-grained semantics of queries and stores [15]. Generative retrieval (GR) has recently emerged as a new paradigm in information retrieval (IR), where the knowledge of the entire corpus is parameterized within a large language model [1,9,21,23,25,28]. GR has achieved state-of-the-art performance across various retrieval tasks, including document retrieval [1,21], code retrieval [11], and industrial applications such as official site retrieval [18] and product retrieval [6,13].

Challenges of Applying GR to store Search. Store search differs significantly from general web search in two key aspects: (i) *From the store perspective*, each store encompasses a rich set of multi-attribute information, which can be broadly categorized into two types: store-specific metadata (e.g., name, brand, description) and product-related information (e.g., product names, categories, descriptions). Enabling a GR model to learn and internalize such diverse information requires a large amount of annotated data, which is often scarce in practice; and (iii) *From the query perspective*, user queries can be *store-centric*, i.e., directly specifying a particular store (e.g., "Apple Store"), or *product-centric*, i.e., describing a product category to find relevant stores (e.g., "sports apparel"). As shown in Fig. 1, in store search, while both require high precision, their retrieval expectations differ: store-centric queries demand exact matches at rank-1, whereas product-centric queries benefit from diverse relevant candidates. Therefore, balancing precision and diversity based on query intent is key to improving user satisfaction. These characteristics make existing GR methods ill-suited for effective store retrieval.

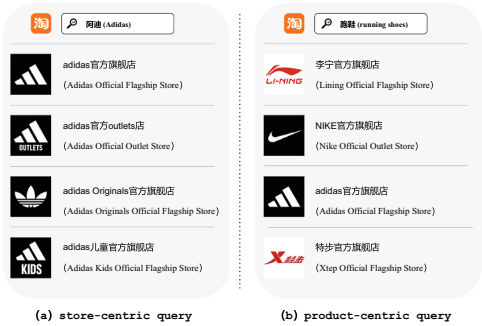

Fig. 1. Schematic diagram of store search in Taobao. (a) Store-centric query; (b) Product-centric query.

Our GR System Tailored for Store Search. Given the challenges of limited annotated data and the need to balance precision and diversity, we propose *GenStore*, a GR framework specifically designed for store search in e-commerce scenarios. GenStore introduces two key contributions:

- **Training with Query Augmentation.** We construct high-quality pseudo-queries from store-level metadata (e.g., store descriptions) and product metadata (e.g., product titles). An LLM-based rewriter generates a diverse set of pseudo-queries from these elements, which are paired with store identifiers for training together with real user queries.
- **Inference with Intent-Aware Decoding.** To balance retrieval precision and diversity, we couple a query-understanding module with an entropy-gated

contrastive decoding algorithm: (i) each query is first processed by an intent classifier and then rewritten according to the detected intent; (ii) the rewritten query is used to perform entropy-gated contrastive decoding under constraints to generate candidate store identifiers, by combining the logits from a fine-tuned GR model (expert) and a smaller general model (amateur), which effectively penalizes repetitive or templated outputs. In addition, we apply an intent-specific diversity penalty to the contrastive scores to further balance the precision and diversity.

Experimental Findings. On a Taobao Store Search benchmark built from all online stores and real queries (Sect. 2), GenStore outperforms dense retrieval and generative retrieval baselines in offline evaluation. In online A/B test, deploying GenStore as an additional recall path in Taobao's store search pipeline yields substantial improvements in key business metrics such as Click-Through Rate and Conversion Rate.

2 Problem Statement

Task Description. We denote the set of all stores as $\mathcal{S} = s_1, \ldots, s_{|\mathcal{S}|}$ and the set of user queries as $\mathcal{Q} = q_1, \ldots, q_{|\mathcal{Q}|}$, where each query q falls into one of two intent categories: (i) *store-centric*, which explicitly names a store or brand, and (ii) *product-centric*, which describes a product category or attribute. Our objective is to build a GR system that, for any query $q \in \mathcal{Q}$, generates the top-k most relevant store identifiers (stids). We require that, for store-centric queries, the top-1 result be highly relevant, and that, for product-centric queries, the candidate set maintain relevance while exhibiting diversity.

Benchmark Construction. We introduce the *Taobao Store Search dataset*, a new dataset specifically designed for the store retrieval task, constructed using real stores and user queries from Taobao[1]:

- *Store metadata* Our corpus consists of all currently registered stores in the Taobao store metadata repository, comprising nearly 9 million stores. For each store, we collect the following metadata: (i) store name; (ii) store main category; (iii) store brand; (iv) store description; and (v) best-selling products.
- *Product metadata* For each store's top-selling products[2], we obtain the following product metadata: (i) product name; (ii) product category; and (iii) product description.
- *User query* We collected 0.95M user query logs for training and evaluation. For training, we pair each query text with its clicked stores as high-quality relevance supervision. For evaluation, since click logs cannot fully cover all relevant stores, we assess retrieval results using relevance models (see Sect. 4.1 for details).

[1] Detailed descriptions are provided in https://anonymous.4open.science/r/Generative-Store-Retrieval-in-Taobao-Search.

[2] The store's top-selling products are determined by ranking items in descending order of their recent sales volumes.

3 Method

3.1 Data Augmentation for Training

We use each store's unique registered name (store name for short) directly as its store identifier (stid). Since store names often contain the store brand, this choice facilitates retrieval for store-centric queries. However, relying solely on the store name makes it challenging to associate an stid with product-centric queries, as the name rarely reflects the store's primary merchandise.

To mitigate this limitation, we introduce diversity-enhanced query augmentation during training. Specifically, we construct five types of augmented data: (i) *Store name variants*: replacing or removing common tokens (e.g., "Shop" and "Official") in the store name; (ii) *Store brands*: the primary brand of the products sold by the store; (iii) *Rewritten store description*: a concise LLM-rewritten summary of the store's business; (iv) *Product categories*: we use store main category from the store metadata as coarse-grained categories (e.g., "Musical Instruments") and the product categories of its top-selling items as fine-grained categories (e.g., "Guitar," "Piano"); and (v) *Representative products*: the store's best-selling products.

We leverage the constructed pairs described above to train our autoregressive retrieval model. Unlike prior GR approaches [1,2,23,24], we do not explicitly separate the indexing and retrieval stages. Instead, our objective is to inject store-specific knowledge into the model through diverse pseudo queries. Given an input query q concatenated with retrieval promptI_r as input, the model is trained to maximize the likelihood of the corresponding stid. Formally, the objective is defined as:

$$\mathcal{L}(\mathcal{S}, \mathcal{A}; \theta) = - \sum_{(a \in \mathcal{A}, s \in \mathcal{S})} \log P(s | I_r, a), \tag{1}$$

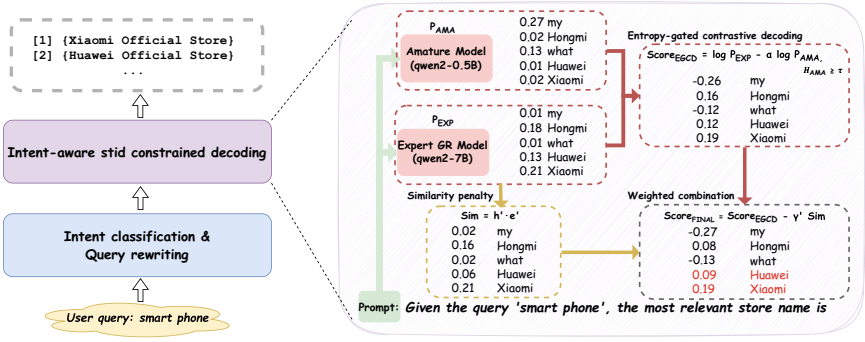

Fig. 2. GenStore's inference pipeline. The left shows the two stage process and the right details our entropy-gated contrastive decoding with similarity penalty. Here, we only illustrate the case where the amateur model's entropy $H_{\text{AMA}} \geq \tau$. For clarity, we let $\alpha = 1$ and $\gamma' = 0.5$.

where θ denotes the parameters of the GR model, and $a \in \mathcal{A}$ can be any form of pseudo queries for the store $s \in \mathcal{S}$ (Fig. 2).

3.2 Intent-Aware Decoding for Inference

Intent Classification and Query Rewriting. During inference, we first classify each test query into its corresponding intent type and then perform unconstrained generation to obtain query rewrites. Specifically, we employ a binary classifier trained on labeled query intents to separate queries into *store-centric* and *product-centric* types. However, due to semantic ambiguity, not all queries can be labeled with high confidence. We therefore apply a confidence-based filtering mechanism that introduces a third label, *uncertain*, whenever the posterior probabilities of the two classes are close.

Next, leveraging the GR model trained via Eq. 1, we perform unconstrained semantic expansion of the user query to exploit store-related knowledge acquired during training and obtain an initial relevance-oriented rewrite. Specifically, we prompt the model to freely generate a set of plausible store mentions the user might intend, without imposing lexical constraints. Although such free-form rewrites may not exactly match the stores' canonical registered names (stids), they help disambiguate intent and serve as a bridge between the query and stids, thereby improving the accuracy of subsequent constrained decoding. For example, given $q =$ "adi", a well-trained GR model may produce "Adidas Direct Store" as a rewrite, even though the canonical name is "Adidas Official Flagship Store".

Intent-Aware Stid Constrained Decoding. Given the rewritten query, we perform constrained decoding via entropy-gated contrastive decoding, with an intent-specific diversity penalty, to generate candidate stids that satisfy the requirements specified in Sect. 2. We next detail these two components.

Entropy-Gated Contrastive Decoding for Precision Improvements. By inspecting the trained model's outputs under constrained decoding, we observe two main issues in GR: (i) the model may generate generic or high-frequency tokens early in decoding, causing the output to diverge from the intended store name; (ii) for particularly difficult queries (e.g., store names composed of meaningless or repetitive strings), the model tends to produce degenerate outputs. Although constrained decoding is applied, the existence of user-submitted store names with repetitive patterns makes such degeneration possible.

To address these issues, we adopt entropy-gated contrastive decoding (EGCD) with a small auxiliary amateur model drawn from the same family as the expert (thus sharing the tokenizer) to suppress generic or templated continuations while preserving expert-plausible tokens. Intuitively, when the amateur is *uncertain* (high-entropy), its preferences are a good proxy for generic trends we wish to penalize; when it is *confident* (low-entropy), contrastive subtraction risks hurting the very correct token we want to promote. We therefore gate the penalty by the amateur's Shannon entropy and additionally enforce a feasibility filter so that tokens the expert considers competitive are never pruned merely

due to contrastive pressure. Formally, at each decoding step, given the expert distribution $P_{\mathrm{EXP}}(\cdot)$, the amateur distribution $P_{\mathrm{AMA}}(\cdot)$, and the amateur entropy H_{AMA}, the single score passed to beam search is

$$\begin{aligned} \mathrm{score}_{\mathrm{EGCD}}(w) = \log P_{\mathrm{EXP}}(w) \; - \; \mathbb{1}\{\, H_{\mathrm{AMA}} \geq \tau \,\}\, \alpha \, \log P_{\mathrm{AMA}}(w) \\ \text{s.t.} \quad \log P_{\mathrm{EXP}}(w) \geq \max_{w'} \log P_{\mathrm{EXP}}(w') + \log \beta, \end{aligned} \tag{2}$$

where $\alpha > 0$ controls the contrastive strength, $\beta \in (0, 1]$ retains only tokens whose expert probability lies within a factor β of the expert's current maximum, and the indicator $\mathbb{1}\{\cdot\}$ activates the contrastive term only when $H_{\mathrm{AMA}} \geq \tau$.

Similarity Penalty Term for Diversity Improvements. To promote diversity without hurting quality, we penalize tokens that are too similar to currently strong beams. At each step, we select a set $\mathcal{B}$ of high-quality beams whose perplexity satisfies $\mathrm{PPL}(b_i) \leq T$ (lower PPL $\Rightarrow$ higher confidence); if $\mathcal{B} = \emptyset$, the penalty is skipped. For each $b_i \in \mathcal{B}$ we take the expert's last hidden state $\mathbf{h}_i$ and normalize $\mathbf{h}'_i = \mathbf{h}_i / \|\mathbf{h}_i\|$, and for each candidate token w we use its normalized embedding $\mathbf{e}'_w = \mathbf{e}_w / \|\mathbf{e}_w\|$. The penalty applied to token w is the maximum cosine similarity to the positives:

$$\mathrm{sim}(w) = \max_{b_i \in \mathcal{B}} \mathbf{h}'_i \cdot \mathbf{e}'_w. \tag{3}$$

Tokens closer to any high-quality path receive larger deductions, nudging beam search toward novel yet plausible continuations.

Weighted Combination for Different Query Intents. We integrate the similarity penalty (Eq. 3) into EGCD (Eq. 2) via an intent-aware weight γ':

$$\mathrm{score}(w) = \mathrm{score}_{\mathrm{EGCD}}(w) - \gamma' \, \mathrm{sim}(w), \quad \gamma' = \begin{cases} 0, & \text{store-centric,} \\ \gamma, & \text{product-centric,} \\ c \cdot \gamma, & \text{uncertain,} \end{cases} \tag{4}$$

where $\gamma \in (0, 1)$ controls penalty strength and c denotes the intent confidence. Beam search then uses $\mathrm{score}(w)$ as the token score at each step, implicitly trading precision for diversity without extra re-ranking.

4 Offline Experiments

4.1 Experimental Setup

Metrics. Because a single store-search query typically corresponds to many relevant stores, human labels are sparse and cannot cover all positives. We therefore evaluate with Taobao's internal e-commerce relevance model (built on a Qwen-13B backbone and trained on 10M human-annotated pairs) which achieves 94.6% agreement with human judgments. We define three metrics based on Taobao's internal relevance scoring system[3]: (i) *Strong Relevance Rate@k (Rel@k):* the

[3] The model assigns a four-level relevance grade, from level 4 (strongest relevance) to level 1 (weakest).

fraction of top-k results whose relevance grade is the highest level; (ii) *Average Relevance@k (Avg. Rel@k):* the mean relevance grade of the top-k results; (iii) *Diversity@k (Div@k):* the number of distinct brands among the highest-graded stores in the top-k. In addition, we evaluate *Hits@20* on a 1k human-labeled set, and employ ChatGPT as a binary relevance judge to complement and cross-check the results of the Taobao relevance model.

Baselines. We compare against term-based retrieval, dense retrieval, and GR baselines: (i) BM25 [16]; (ii) DPR [3]; (iii) DSI [21]; (iv) SEAL [1]; and (v) MINDER [8]. We do not include Taobao's production system in offline comparisons, as it is a multi-stage pipeline (recall, pre-ranking, fine-ranking) and thus not directly comparable to the single-stage baselines above. Instead, we provide a head-to-head comparison via online A/B tests against the production system.

Implementation Details. All GR models are implemented in PyTorch and fine-tuned from the Qwen2-7B backbone [22] with AdamW [10]. At inference, we use beam search (beam=20). For EGCD, GenStore adopts Qwen2-0.5B as the amateur model. Decoding hyperparameters are α=0.1, β=0.05, γ=0.05, τ=3, and T=1.1. Experiments run on 2× NVIDIA H20 GPUs.

Table 1. Comparison of retrieval performance across GenStore and baselines.

	Method	Rel@1	Rel@20	Avg. Rel@1	Avg. Rel@20	Div@20
Term-based/Rense	BM25	5.6	4.7	1.49	1.44	0.65
	DPR	14.8	10.0	1.76	1.70	1.44
Generative	DSI	14.2	10.3	1.77	1.62	1.24
	SEAL	<u>17.8</u>	<u>12.6</u>	<u>1.92</u>	<u>1.74</u>	**1.71**
	MINDER	14.8	12.0	1.77	1.67	1.36
Ours	**GenStore**	**20.6**	**14.2**	**1.97**	**1.76**	<u>1.38</u>

4.2 Experimental Results

Main results. As shown in Table 1, GenStore significantly outperforms all baselines on the real-world store search dataset. Specifically, we have the following observations: (i) GenStore achieves the highest top-1 relevance (Rel@1), demonstrating its ability to surface the most pertinent store at rank one; (ii) it leads across all overall relevance metrics (Rel@20, Avg. Rel@1, Avg. Rel@20), evidencing superior end-to-end retrieval effectiveness; (iii) its diversity score (Div@20) trails SEAL by only a small margin, indicating that GenStore maintains maximal brand variety while preserving high precision; and (iv) GR methods that construct stids from text (SEAL, MINDER, and our GenStore) all markedly outperform both term-based and dense retrieval baselines, underscoring the strong potential of text-based stid methods in store search.

To complement and cross-check the relevance-model results, we evaluate *Hits@20* on a 1k human-labeled set and employ ChatGPT as a binary relevance judge and the results are shown in Table 2. Compared with Table 1, the relative ordering of methods is largely consistent across the relevance model, ChatGPT, and human labels, supporting the effectiveness of *GenStore*. However, absolute scores on the human-labeled set are uniformly low, reflecting limited coverage (a single query often has many relevant stores). Given the high agreement between ChatGPT and the production relevance model, we adopt the relevance model as the primary offline evaluator in subsequent experiments.

Table 2. Human-labeled evaluation and GPT-based evaluation.

Method	Hits@20	Rel@1 (GPT)
BM25	3.3%	48%
DPR	6.6%	65%
DSI	4.6%	55%
SEAL	5.8%	67%
MINDER	5.2%	63%
GenStore	**8.5%**	**72%**

Ablation study. The ablation results are shown in Table 3. We can draw the following conclusions: (i) data augmentation meaningfully boosts overall relevance, demonstrating its role in enriching the model's exposure to diverse store attributes; (ii) query rewriting is critical for GenStore, as its removal leads to a substantial drop in both relevance and diversity; (iii) entropy-gated contrastive decoding effectively suppresses degenerate outputs and contributes to a balanced improvement in precision and result variety; and (iv) the similarity penalty refines the precision-diversity trade-off and enhances diversity. Therefore only the full GenStore could achieves the best trade-off between high precision and competitive diversity.

Table 3. Ablation study of GenStore's core components.

Variants	Rel@1	Avg. Rel@20	Div@20
GenStore	**20.6**	1.76	**1.38**
w/o data augmentation	18.7	1.72	1.35
w/o query rewriting	16.0	1.65	0.97
w/o contrastive decoding	17.5	1.68	1.23
w/o similarity penalty	20.3	**1.78**	1.21

Efficiency. We evaluate the time overhead of adding an amateur model for EGCD and a similarity penalty to GenStore by comparing average retrieval latency under identical hardware. Our results show that GenStore is only 46% slower than standard GR. Moreover, because the expert and amateur models decode independently, their computations can be parallelized, further reducing this overhead in practice.

Performance Under Different Query Intents. Table 4 summarizes GenStore's effectiveness across the three intent groups. We draw the following insights: (i) for store-centric queries, GenStore achieves the highest Rel@1, markedly outperforming all baselines and demonstrating superior capability in exact retrieval; (ii) across all three intent categories, GenStore consistently attains the top Avg. Rel@20 scores, indicating its robustness in maintaining high relevance throughout the result list; (iii) in product-centric queries, GenStore's Div@20 surpasses most baselines and approaches the highest diversity levels while still preserving relevance; and (iv) GenStore significantly improves both Rel@1 and Div@20 for uncertain queries, confirming its ability to balance precision and diversity when intent is ambiguous. The consistent superiority of GenStore across all intents validates the effectiveness of combining intent-aware query understanding with entropy-gated contrastive decoding for a unified GR framework.

Table 4. Retrieval performance by query intent (store-centric: 46.0%, product-centric: 53.1%, uncertain: 0.9%).

Method	Rel@1	Avg. Rel@20	Div@20
Store-centric			
BM25	6.5	1.36	–
DPR	5.9	1.63	–
DSI	11.8	1.71	–
SEAL	15.2	<u>1.83</u>	–
MINDER	<u>16.0</u>	1.80	–
GenStore	**24.2**	**1.97**	–
Product-centric			
BM25	–	1.48	0.55
DPR	–	<u>1.66</u>	**1.91**
DSI	–	1.55	1.25
SEAL	–	1.63	1.46
MINDER	–	1.54	1.23
GenStore	–	**1.76**	<u>1.79</u>
Uncertain			
BM25	0.0	1.50	0.06
DPR	6.3	<u>1.53</u>	<u>1.29</u>
DSI	<u>13.3</u>	1.17	0.24
SEAL	12.5	<u>1.53</u>	1.06
MINDER	0.0	1.29	0.65
GenStore	**20.0**	**1.58**	**1.53**

Fig. 3. Case study. We selected one example each for store-centric and product-centric queries to compare GenStore against the next-best method, SEAL, showing only the top-3 retrieved stores. Green boxes indicate perfectly relevant stores, red boxes denote completely irrelevant stores, and purple boxes mark relevant stores that are not the user's first preference. (Color figure online)

Case Study. Figure 3 contrasts GenStore with SEAL. GenStore returns more relevant results for both query types, while SEAL tends to repeat patterns or match only at the token level. However, in the product-centric case, GenStore still includes a "Solid State Drive SSD" store (a token-level match). This likely stems from the LLM's strong lexical-association ability and its limited understanding of ranking.

5 Online Test

To validate GenStore in real-world conditions, we conducted an online A/B test.

Online Deployment. Figure 4 illustrates how GenStore integrates query understanding, recall, and pre-ranking within the system. The Taobao online store search pipeline is divided into two primary stages: recall and ranking. The recall stage already incorporates multiple retrieval branches including term-based methods such as BM25 and dense-vector methods like DPR. In this work, we add GenStore as an additional recall path. To mitigate latency, we precompute and cache results for head queries (those issued more than 100 times in the past 30 days), thereby reducing on-the-fly inferences. Our cache contains approximately 3.12 million entries and covers 87.2% of query traffic.

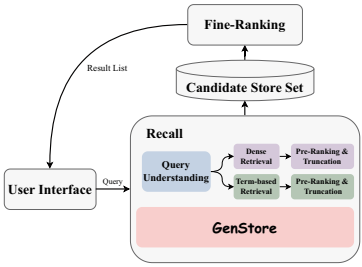

Fig. 4. The deployment of GenStore and the role of retrieval plays in the Taobao store search engine.

Online Metrics. We use several key business metrics: (i) *Click-Through Rate (CTR):* the ratio of clicks to impressions; (ii) *Conversion Rate (CVR):* the ratio of completed transactions to impressions; (iii) *Click Count (CC):* the total number of clicks received; (iv) *Order Count (OC):* the total number of completed transactions; (v) *Gross Merchandise Value (GMV):* the total monetary value of completed transactions; (vi) *Relevance (Rel):* the average relevance score of exposed results, as judged by human assessors; and (vii) *Page View Rate (PVR):* the proportion of exposures attributable to each individual recall path.

Online Performance. As shown in Table 5, we can conclude that: (i) GenStore increases user engagement, as evidenced by higher CC and OC in live traffic; (ii) the overall transaction volume, measured by GMV, benefits from the improved recall quality of GenStore; (iii) precision gains at rank one and across all result positions confirm that GenStore's relevance enhancements

Table 5. Online A/B test results.

CC	OC	GMV	Rel@1	Rel	Div
+1.73%	+0.64%	+1.25%	+1.21%	+1.97%	+0.97%

translate into real-world effectiveness; and (iv) introducing GenStore as an additional recall path brings more varied store options to users without sacrificing accuracy.

To understand GenStore's unique contribution, we compared different recall paths in the enhanced system, as shown in Table 6. We can conclude that GenStore exhibits a lower overall exposure rate compared to traditional recall paths, indicating it retrieves fewer stores in total. Despite this, GenStore achieves the highest exclusive CTR and CVR, demonstrating its ability to surface unique, high-quality stores that other methods do not recall.

Table 6. Performance of different recall paths. Ov. denotes overall and Ex. denotes exclusive.

Recall Path	Ov. PVR	Ex. PVR	Ex. CTR	Ex. CVR
BM25	**45.4**	**5.3**	3.9	1.6
Dense	35.5	1.2	4.0	1.7
GenStore	16.7	3.2	**4.3**	**1.8**

6 Related Work

Generative Retrieval (GR). Numerous studies have explored GR for standard document retrieval, including the design of document identifiers (docids) [8,21,26], training pipelines [9,19], and constrained decoding mechanisms [1,26]. Recently, GR has been extended to other IR domains, such as code retrieval [11], image retrieval [27], and book retrieval [20]. However, these methods are tailored to the characteristics of their respective domains and do not readily generalize to the store retrieval scenario. This work addresses that gap.

Contrastive Decoding (CD). In recent years, CD has emerged as an effective way to curb degenerative LLM outputs by injecting negative signals alongside the model's own probabilities. Classical CD [17] penalizes tokens overly similar to prior context, preserving semantics while improving diversity; speculative CD [7, 12] uses a smaller "amateur" to propose negatives for the expert; adaptive CD [5] modulates the penalty by context reliability; and distillation CD [14] synthesizes negatives via quantization/dropout when no auxiliary model is available.

7 Conclusion

We proposed **GenStore**, a generative retrieval framework for e-commerce store search. By training with diverse pseudo-queries and decoding with entropy-gated contrast plus intent-aware diversity control, GenStore directly generates store identifiers and achieves consistent gains in relevance and diversity in both offline and online A/B evaluations. However, EGCD adds inference overhead and may underperform when expert–amateur predictions coincide, and our study focuses on store retrieval only. Our future work will explore lighter contrastive gates and extensions to product- and multi-vertical retrieval.

Acknowledgments. This work was funded by the National Natural Science Foundation of China under Grants No. 62472408, U25B2076 and 62441229, the Strategic Priority Research Program of the CAS under Grant No. XDB0680102, 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 agreements No. 101070212 (FINDHR) and 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

1. Bevilacqua, M., Ottaviano, G., Lewis, P., Yih, S., Riedel, S., Petroni, F.: Autoregressive search engines: Generating substrings as document identifiers. Adv. Neural. Inf. Process. Syst. **35**, 31668–31683 (2022)
2. Chen, J., Zhang, R., Guo, J., Fan, Y., Cheng, X.: Gere: Generative evidence retrieval for fact verification. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2184–2189 (2022)
3. Karpukhin, V., Oguz, B., Min, S., Lewis, P.S., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: EMNLP (1), pp. 6769–6781 (2020)
4. Khattab, O., Zaharia, M.: 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, pp. 39–48 (2020)
5. Kim, Y., et al.: Adaptive contrastive decoding in retrieval-augmented generation for handling noisy contexts. arXiv preprint arXiv:2408.01084 (2024)
6. Li, M., Wang, H., Chen, Z., Nie, G., Qiu, Y., Tang, G., Liu, L., Zhuo, J.: Generative retrieval with preference optimization for e-commerce search. arXiv preprint arXiv:2407.19829 (2024)
7. Li, X.L., et al.: Contrastive decoding: Open-ended text generation as optimization
8. Li, Y., Yang, N., Wang, L., Wei, F., Li, W.: Multiview identifiers enhanced generative retrieval. arXiv preprint arXiv:2305.16675 (2023)
9. Li, Y., Yang, N., Wang, L., Wei, F., Li, W.: Learning to rank in generative retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 8716–8723 (2024)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
11. Nadeem, U., Ziems, N., Wu, S.: Codedsi: Differentiable code search. arXiv preprint arXiv:2210.00328 (2022)
12. O'Brien, S., Lewis, M.: Contrastive decoding improves reasoning in large language models. arXiv preprint arXiv:2309.09117 (2023)
13. Pang, M., et al.: Generative retrieval and alignment model: A new paradigm for e-commerce retrieval. In: Companion Proceedings of the ACM on Web Conference 2025. pp. 413–421 (2025)
14. Phan, P., Tran, H., Phan, L.: Distillation contrastive decoding: Improving llms reasoning with contrastive decoding and distillation. arXiv preprint arXiv:2402.14874 (2024)

15. Ren, Z., He, X., Yin, D., de Rijke, M.: Information discovery in e-commerce. Found. Trends Inf. Retr. **18**(3–4), 262–535 (2024)
16. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009)
17. Su, Y., Collier, N.: Contrastive search is what you need for neural text generation. arXiv preprint arXiv:2210.14140 (2022)
18. Tang, Y., Zhang, R., Guo, J., Chen, J., Zhu, Z., Wang, S., Yin, D., Cheng, X.: Semantic-enhanced differentiable search index inspired by learning strategies. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4904–4913 (2023)
19. Tang, Y., Zhang, R., Guo, J., De Rijke, M., Chen, W., Cheng, X.: Listwise generative retrieval models via a sequential learning process. ACM Trans. Inf. Syst. **42**(5), 1–31 (2024)
20. Tang, Y., et al.: Generative retrieval for book search. arXiv preprint arXiv:2501.11034 (2025)
21. Tay, Y., et al.: Transformer memory as a differentiable search index. Adv. Neural. Inf. Process. Syst. **35**, 21831–21843 (2022)
22. Team, Q.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
23. Wang, Y., Hou, Y., Wang, H., Miao, Z., Wu, S., Chen, Q., Xia, Y., Chi, C., Zhao, G., Liu, Z., et al.: A neural corpus indexer for document retrieval. Adv. Neural. Inf. Process. Syst. **35**, 25600–25614 (2022)
24. Zeng, H., Luo, C., Jin, B., Sarwar, S.M., Wei, T., Zamani, H.: Scalable and effective generative information retrieval. In: Proceedings of the ACM Web Conference 2024, pp. 1441–1452 (2024)
25. Zeng, H., Luo, C., Zamani, H.: 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, pp. 469–480 (2024)
26. Zhang, P., Liu, Z., Zhou, Y., Dou, Z., Liu, F., Cao, Z.: Generative retrieval via term set generation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 458–468 (2024)
27. Zhang, Y., et al.: Irgen: generative modeling for image retrieval. In: European Conference on Computer Vision, pp. 21–41. Springer (2024)
28. Zhuang, S., et al.: Bridging the gap between indexing and retrieval for differentiable search index with query generation. arXiv preprint arXiv:2206.10128 (2022)