

TSGR: Taobao Search Generative Retrieval

Tianyu Zhan*

Zhejiang University
Hangzhou, China
yuzt@zju.edu.cn

Gui Ling*

Taobao & Tmall Group of Alibaba
Hangzhou, China
linggui.lg@taobao.com

Tong Xiong*

Taobao & Tmall Group of Alibaba
Hangzhou, China
xiongtong.xt@taobao.com

Kunhai Lin*

Taobao & Tmall Group of Alibaba
Hangzhou, China
linkunhai.lkh@taobao.com

Yang Wang

Taobao & Tmall Group of Alibaba
Hangzhou, China
wy416570@taobao.com

Kaixuan Zhang

Taobao & Tmall Group of Alibaba
Hangzhou, China
zhangkaixuan.zkx@taobao.com

Zhihong Chen

Taobao & Tmall Group of Alibaba
Hangzhou, China
jhon.czh@taobao.com

Yuliang Yan[†]

Taobao & Tmall Group of Alibaba
Hangzhou, China
yuliang.yyl@taobao.com

Dan Ou

Taobao & Tmall Group of Alibaba
Hangzhou, China
oudan.od@taobao.com

Shengyu Zhang[†]

Zhejiang University
Hangzhou, China
sy_zhang@zju.edu.cn

Haihong Tang

Taobao & Tmall Group of Alibaba
Hangzhou, China
piaoxue@taobao.com

Bo Zheng

Taobao & Tmall Group of Alibaba
Hangzhou, China
bozheng@taobao.com

Abstract

Generative retrieval (GR) has demonstrated strong promise for industrial e-commerce search by training a single autoregressive model to directly generate the Semantic IDs (SIDs) of target items. However, existing GR systems are primarily optimized for semantic matching and remain insensitive to item business value: SID construction is value-unaware, and candidates are ranked without access to item side-info. Consequently, high-value items are often missed or deprioritized at the retrieval stage, limiting downstream business impact. This limitation is particularly critical in industrial settings such as Taobao Search, where business objectives are central to system design. To address this, we propose **Taobao Search Generative Retrieval (TSGR)**, a unified generative retrieval framework that incorporates value awareness into both item representation and candidate ranking. 1) For item representation, TSGR introduces **Query-aware Parallel SID (QP-SID)**, which encodes query-conditioned value orderings into the SID construction by building parallel codebooks derived from query-item statistics, so that higher-value and query-relevant items are assigned better token indices. 2) For candidate ranking, we introduce a **Value-aware Ranking Module (VRM)** that is built upon and jointly optimized with the GR, enabling a single model to seamlessly serve as both

retriever and pre-ranker without a dedicated pre-ranking stage. A progressive training pipeline further aligns the model with semantic relevance, user preferences, and business objectives. Offline experiments show that TSGR achieves an 9.16% improvement in HR@1000, and online A/B tests further validate its effectiveness, yielding gains of +0.43% in IPV, +1.12% in Transaction Count, and +1.64% in GMV.

CCS Concepts

• **Information systems** → **Retrieval models and ranking.**

Keywords

Generative Retrieval, Semantic ID, Business Value Modeling

ACM Reference Format:

Tianyu Zhan, Gui Ling, Tong Xiong, Kunhai Lin, Yang Wang, Kaixuan Zhang, Zhihong Chen, Yuliang Yan, Dan Ou, Shengyu Zhang, Haihong Tang, and Bo Zheng. 2018. TSGR: Taobao Search Generative Retrieval. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Generative retrieval (GR) has emerged as a promising paradigm for the retrieval stage of industrial e-commerce search, leveraging the sequence generation capability of LLMs to train a single autoregressive model that directly generates the Semantic IDs (SIDs) of target items [17, 22]. This formulation enables end-to-end optimization while considerably reducing the infrastructure overhead of maintaining large-scale indices. Building on these advantages, GR has been rapidly deployed at scale across a range of industrial applications, including recommendation [34], search [1], and advertising [29, 31].

*Equal contribution. Work is done during Tianyu Zhan's internship at Alibaba Group.

[†]Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXXX.XXXXXXX>

Beyond these practical benefits, GR is conceptually well aligned with the nature of information retrieval: the generative model encodes knowledge of the entire item collection into its parameters and reconstructs relevant items in response to a query, while SID encoding compresses the vast product space into a compact token vocabulary, making full-space optimization feasible and alleviating the negative sampling bias in conventional dual-encoder models. Together, these properties make GR a strong foundation for large-scale semantic matching in industrial search.

However, industrial e-commerce search is ultimately driven by business objectives (e.g., CTR, CVR and GMV). Downstream ranking stages rely heavily on these signals to score candidates, precisely because two items equally relevant to a query may differ markedly in conversion probability. When the retrieval stage is unaware of value signals, it may retrieve semantically reasonable but commercially inferior candidates, limiting the business impact of downstream rankers [31]. This insensitivity to business value stems from how existing e-commerce GR frameworks operate, where value signals are seldom incorporated into either item representation or model output, giving rise to two key limitations.

Value-insensitive SID construction. Existing RQ-based SID pipelines are primarily designed to preserve multimodal semantic similarity and generally lack an explicit mechanism to reflect item business value [17, 34]. As a result, items of similar semantics but markedly different value tend to be mapped to the same SID neighborhood, yielding a retrieval space that is semantically organized but largely value-undifferentiated. Moreover, SIDs are typically fixed globally and remain query-agnostic, so that each item carries an identical identifier regardless of query, overlooking how its relevance and value can vary across different search intents.

Value-unaware candidate ranking. Standard GR ranks candidates by the probability of SID tokens under beam search, reflecting semantic relevance more than business relevance. Furthermore, since candidates are generated dynamically at inference time, the model cannot prefetch rich item side-info at input (e.g., features and value signals). Consequently, the candidates ranked by beam search score often diverge from the items most likely to be clicked or purchased.

To address these limitations, we propose Taobao Search Generative Retrieval (**TSGR**), a unified generative framework that incorporates value awareness into both item representation and candidate ranking via two core innovations. For item representation, we observe that earlier token indices in the last-level SID codebook are mapped to by more items and thus receive more training signal. Motivated by this, we propose **Query-aware Parallel SID (QP-SID)**, which constructs multiple parallel orderings of items, including a default ordering based on overall clicks and query-conditioned orderings based on query-specific click statistics. By assigning higher-value and query-relevant items to earlier indices, QP-SID naturally aligns generation likelihood with both business value and query relevance. For candidate ranking, we introduce a **Value-aware Ranking Module (VRM)** that extracts user representations from the hidden states of the generative backbone and fuses them with rich item side-info via cross-attention, producing scores aligned with business objectives for re-ranking beam search candidates. Crucially, this module is jointly optimized with the generation objective within the same model, naturally extending GR into a unified retrieval and

pre-ranking framework without a dedicated pre-ranking stage. To further align TSGR with downstream business objectives, we adopt a progressive training pipeline consisting of Pre-SFT multi-task pretraining and SFT with weighted multi-positive supervision. We also explore the role of reinforcement learning (RL), and find that applying the VRM makes superior performance.

Our main contributions are summarized as follows:

- We propose **TSGR**, a unified generative framework that integrates retrieval and pre-ranking within a single model and aligns the full pipeline with business objectives.
- We design **QP-SID**, an item representation that encodes value signals into SIDs and derives query-aware orderings from query-item click statistics, addressing the value-insensitivity and query-agnosticism of conventional SIDs.
- We introduce the **VRM** that fuses backbone-derived user representations with item side-info via cross-attention, and jointly optimize it with the generation backbone so that a single model serves as both retriever and pre-ranker.
- We conduct extensive offline experiments on real-world Taobao search data and large-scale online A/B tests, yielding gains of +0.43% in IPV, +1.12% in Transaction Count, and +1.64% in GMV.

2 Related Work

2.1 Generative Retrieval and Recommendation

Generative retrieval and recommendation reframe item retrieval as a sequence generation problem, in which a model directly produces the discrete SIDs of items in an autoregressive manner. DSI [22] first demonstrates this paradigm by memorizing a document corpus within model parameters. TIGER [17] employs RQ-VAE to compress item embeddings into hierarchical token sequences, unifying semantic and collaborative signals through a shared quantized vocabulary, while IDGenRec [21] further connects generative models with item ID spaces via textual ID generation.

As generative models scale to billion-parameter LLMs, this paradigm has evolved from proof-of-concept to industrial deployment. On the architectural side, HSTU [28] and HLLM [3] improve efficiency through tailored attention mechanisms and hierarchical LLM designs, whereas OneRec [34] casts recommendation as a generic sequence prediction task with preference alignment. In e-commerce search, OneSearch [1] proposes a unified end-to-end generative framework, which OneSearch-V2 [2] extends with latent reasoning and self-distillation. Similar trends have also emerged in advertising [29, 31, 33] and general search [5, 6].

2.2 Semantic ID Construction

Semantic IDs convert items into compact discrete token sequences that serve as generation targets for generative retrieval. TIGER [17] pioneers RQ-VAE for hierarchical quantization, and subsequent works simplify this process via RQ-KMeans [34], OPQ [7], and joint product quantization [16]. To enrich SID semantics, LETTER [23] and LC-Rec [32] align multimodal features with collaborative signals, whereas GSID [27], CAT-ID2 [12], and Hi-Gen [25] inject domain-specific structures such as attribute hierarchies and category trees. CQ-SID [35] encodes items into semantic clusters rather than unique identifiers to reduce beam search cost. At industrial

scale, FORGE [4] benchmarks SID construction on Taobao with 14 billion interactions, and UniSID [9] jointly optimizes SID construction with the recommendation objective in an end-to-end manner.

All existing methods assign each item a single global SID, confining it to a fixed representation regardless of query context and overlooking item business value during codebook construction. TSGR addresses both limitations by constructing parallel SIDs per item via query-aware, value-informed codebooks, allowing the same item to be associated with different representations across queries while prioritizing high-value items within each cluster.

2.3 Value-aware Ranking

Industrial search systems typically adopt a cascaded architecture in which retrieval and ranking are handled by separate models [15], which tends to introduce objective misalignment and prevents item value signals from participating in candidate generation [24]. To narrow this gap, a growing line of work explores using generative models directly for ranking. GenRank [8] demonstrates generative ranking at industrial scale, and QGS [20] integrates heterogeneous features into generative ranking through HFG-Attention. UniVA [31] introduces a Generation-as-Ranking formulation with eCPM-aware reinforcement learning, while GoalRank [30], GReF [10], and DeGRe [19] incorporate ranking signals through reward models and dense supervision.

However, these methods largely rely on generation probabilities as ranking scores, which reflect sequence likelihood rather than actual item value, offering limited support for the rich item side-info that are critical for value estimation in industrial systems. In contrast, TSGR directly incorporates item side-info into value-aware scoring within the generative model, unifying retrieval and ranking under a single jointly optimized objective.

3 Methodology

TSGR is a unified generative framework for industrial e-commerce search that integrates retrieval and pre-ranking within a single model. As illustrated in Figure 1, the **Query-aware Parallel SID Construction** (Section 3.1) builds distinct SID paths for each item and selects the most appropriate one conditioned on the query. The **Value-aware Ranking Module** (Section 3.2) is built upon the generative backbone and re-ranks beam search candidates. Section 3.3 presents a training pipeline that progressively develops model capability through Pre-SFT multi-task pretraining and SFT with weighted multi-positive training. Besides, We also explore the role of RL.

3.1 Query-aware Parallel SID Construction

We propose a parallel SID construction method that builds distinct SID representations for each item based on query-item statistics. Specifically, we leverage two types of prior information to construct two codebooks: a semantic codebook configured as 32768×8192 to capture the semantic structure of items, and an efficiency codebook of overall size 8192 composed of N parallel branches, enabling diverse retrieval paths while concentrating the decoding budget on high-probability branches and avoiding unnecessary search capacity on low-probability paths [11].

Semantic Codebook Construction. The first two levels form the semantic codebook, which is built upon existing item features such as same-item cluster and item category, to construct a natural item hierarchy $Item \in Cluster \in Category$. We first obtain cluster-level and category-level embeddings by averaging the embeddings of all items within each group:

$$\mathbf{e}_{\text{cat}}(g) = \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} \mathbf{e}_i \quad \mathbf{e}_{\text{cluster}}(g) = \frac{1}{|C|} \sum_{i \in C} \mathbf{e}_i \quad (1)$$

where $\mathcal{G}$ and C denote items in the same category and cluster, and $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding of item i . The Level-1 codebook is initialized directly with category embeddings, as the number of categories (8100+) closely matches the codebook size of 8192. RQ-KMeans is then applied to the residuals between cluster and category embeddings to learn the Level-2 codebook:

$$\mathbf{r}(g) = \mathbf{e}_{\text{cluster}}(g) - \mathbf{e}_{\text{cat}}(g) \quad (2)$$

This produces a three-level $8192 \times 4 \times 8192$ intermediate codebook. Since a four-level codebook in total would incur prohibitive deployment cost in production, we merge the first two levels into a single level, resulting in the final semantic codebook of size 32768×8192 .

This construction injects two structural priors into the semantic codebook:

- *Category prior*: the domain taxonomy is embedded directly into the coarsest partition, so that the SID space inherits a meaningful category structure.
- *Cluster prior*: by quantizing cluster centroids rather than individual item embeddings, all items within the same cluster are guaranteed to share an identical semantic SID prefix.

Together, these priors make the SID space semantically well-organized and provide a clean cluster partition on which the efficiency codebook is built.

Efficiency Codebook Construction. The third level constitutes the efficiency codebook, constructed independently within each cluster by ordering items according to their click-based value. Higher-value items occupy earlier token indices, which are visited more frequently during training, concentrating the decoding budget on the most promising items and aligning generation likelihood with item value.

A straightforward design ranks items by their click count over the past 30 days, giving the default ordering:

$$\pi_0 = \text{argsort}_{i \in C} s_0(i) \quad (3)$$

where C denotes the set of items in the cluster and $s_0(i)$ is the click count of item i . However, this ordering remains query-agnostic, overlooking the fact that an item may be highly relevant to one query but not another. To overcome this, we propose **QP-SID**, built on the observation that each query can be segmented into key terms and each item belongs to one cluster, so term-cluster relations can be inferred from query-item statistics. Concretely, for each cluster we identify the top-3 most representative terms, filtering out those whose coverage exceeds 80%, where coverage is defined as the fraction of items in the cluster clicked under the term:

$$\text{cov}(t, C) = \frac{|\{i \in C : s(i, t) > 0\}|}{|C|} \quad (4)$$

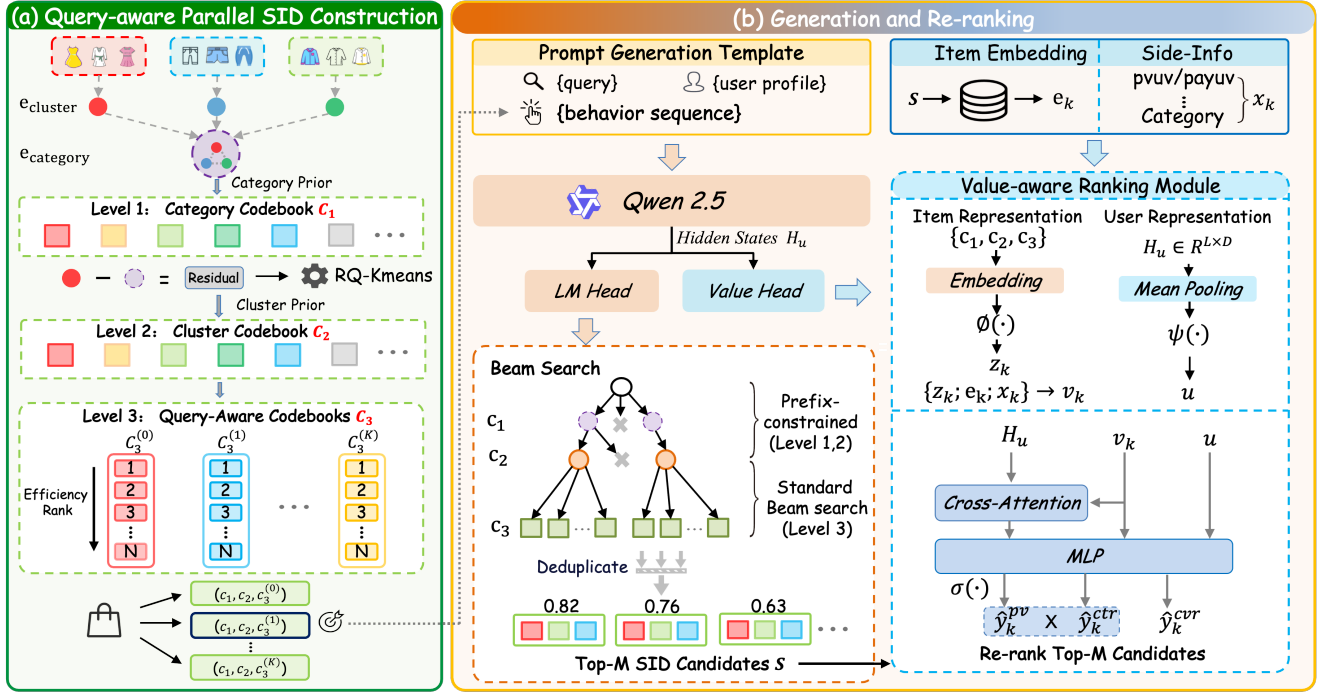


Figure 1: Overview of TSGR, a unified generative framework that jointly handles retrieval and pre-ranking within a single model. (a) Parallel SID Construction builds a three-level codebook using category and cluster priors at Levels 1–2, and query-aware codebooks at Level 3. (b) Generative Backbone performs hierarchical beam search with prefix-constrained decoding at Levels 1–2 and standard beam search at Level 3. VRM re-ranks candidates by fusing SID embeddings with item side-info.

A term with near-complete cluster coverage carries little discriminative signal, as it is associated with virtually all items regardless of query intent. For each retained representative term t , let $s(i, t)$ denote the click count of item i under term t over the past 30 days. Ranking items by $s(i, t)$ produces the term-specific ordering:

$$\pi_t = \text{argsort}_{i \in C} s(i, t) \quad (5)$$

This produces 3 term-conditioned orderings together with the default ordering π_0 , giving 4 parallel orderings per cluster. The case study can be found in Appendix A.4.

3.2 Value-aware Ranking Module

In GR, candidates are ranked by generation probability, which reflects sequence likelihood rather than business value. Since candidate items are only known after decoding, their side-info cannot be incorporated during generation. A natural remedy is a separate downstream pre-ranking model, but this couples two models with misaligned objectives: the GR model optimizes SID generation likelihood over its own retrieval distribution, while the pre-ranking model scores candidates from the full multi-source pool, introducing a distribution gap that undermines their combination.

To this end, we introduce the **VRM** on top of the generative backbone. As illustrated in Figure 1(b), the module takes the user representations of the generative model together with item side-info as input, and outputs a value score for each candidate. We

describe its four components in turn: item representation, user representation, user-item interaction and candidates scoring.

Item Representation. For each candidate $(c_1, c_2, c_3) \in \mathcal{S}$ from beam search, we construct the item representation from three complementary sources. The first is an item side-info vector $\mathbf{x}_k \in \mathbb{R}^{d_f}$ retrieved from offline feature tables, which aggregates four categories of features: (1) behavioral statistics over multiple time windows (e.g., page views and payment counts); (2) category features derived from the item’s category hierarchy; (3) global seller statistics that capture cross-scenario behavioral patterns; and (4) query-item co-occurrence features. The second is an item embedding $\mathbf{e}_k$ retrieved from the offline feature table, which provides information that may be lost during SID construction. The third is a SID embedding, obtained by mapping the candidate SIDs (c_1, c_2, c_3) through the backbone’s embedding layer and compressing it with an MLP $\phi(\cdot)$:

$$\mathbf{z}_k = \phi(\text{Emb}(c_1), \text{Emb}(c_2), \text{Emb}(c_3)) \quad (6)$$

The three are concatenated to form the final item representation:

$$\mathbf{v}_k = [\mathbf{x}_k; \mathbf{e}_k; \mathbf{z}_k] \quad (7)$$

User Representation. The user representation reuses the hidden states of the generative backbone, thus incurring no additional encoding cost. Given the input prompt of length L , the backbone produces contextualized representations $H_u \in \mathbb{R}^{L \times D}$, where D is the hidden dimension. A global user representation is obtained by

Table 1: Pre-SFT task design. Each task activates a specific capability dimension by varying the input context. Most tasks share a unified SID output format; the title generation task uses natural language output to activate intent understanding at the textual level.

Task Category	Input	Output	Capability
Encoding	Item title	SID	Learn item semantics to SID space mapping
Personalization	User profile + query + behavior seq.	Title	Activate user intent understanding at textual level
Retrieval	Search query	SID	Establish query–item relevance for in-domain retrieval
Collaboration	Trigger item (title + SID)	SID	Inject co-purchase collaborative signals into SID space
Recommendation	User profile + behavior seq.	SID	Enable personalization and cross-domain generalization

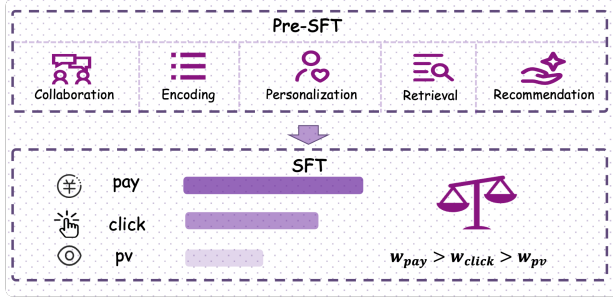


Figure 2: Training Pipeline. Model capabilities are developed progressively via Pre-SFT multi-task pretraining and SFT with a weighted multi-positive loss.

mean pooling over all token positions followed by an MLP $\psi(\cdot)$:

$$\mathbf{u} = \psi\left(\frac{1}{L} \sum_{i=1}^L \mathbf{H}_u[i]\right) \quad (8)$$

User-Item Interaction. To capture fine-grained relevance beyond the global summary, we introduce an attention module in which the item representation $\mathbf{v}_k$ attends to the user context representations $\mathbf{H}_u$:

$$\mathbf{a}_k = \text{CrossAttn}(\mathbf{v}_k, \mathbf{H}_u, \mathbf{H}_u) \quad (9)$$

where $\mathbf{v}_k$ serves as the query and $\mathbf{H}_u$ serves as both the key and the value. This allows each candidate to selectively attend to the most relevant parts of the user context, yielding an interaction representation $\mathbf{a}_k$ that encodes user-item relevance.

Feature Fusion and Scoring. The interaction representation, the user representation, and the item representation are concatenated and fed into three parallel heads to produce corresponding prediction scores:

$$\hat{y}_k^t = \sigma(\text{MLP}_t([\mathbf{a}_k; \mathbf{u}; \mathbf{v}_k])) \quad t \in \{\text{pv}, \text{ctr}, \text{cvr}\} \quad (10)$$

where $\sigma(\cdot)$ is the sigmoid function. At inference, candidates are re-ranked by the composite score $\hat{y}_k^{\text{pv}} \times \hat{y}_k^{\text{ctr}}$ to produce the final output. Since this module shares the backbone embedding and reuses the hidden states computed during generation, it unifies retrieval and value-aware ranking within a single model at negligible additional cost.

3.3 Training Pipeline

Training TSGR requires three capabilities to be developed together: semantic alignment between item content and the SID space, personalized query-to-SID generation, and value-aware candidate ranking. Acquiring all three within a single supervised stage is difficult, so we adopt a pipeline that builds them progressively as shown in Figure 2, with detailed prompt templates provided in Appendix A.5.

3.3.1 Pre-SFT. Pretrained LLMs are not equipped to map items or queries to discrete SID tokens. Instead of forcing the model to acquire this and all other capabilities at once, Pre-SFT decomposes the target objective into simpler sub-tasks, each focusing on a single capability dimension. Training on large-scale samples across these sub-tasks activates the necessary capabilities in advance, which lowers both the learning difficulty and the data requirements of the subsequent SFT stage.

We accordingly design five categories of Pre-SFT tasks, summarized in Table 1. Most tasks share a unified SID output format but differ in input context. During training, all tasks are shuffled and optimized jointly with the standard cross-entropy loss.

3.3.2 SFT. The SFT stage adapts the Pre-SFT-initialized model to the personalized retrieval task.

Prompt Generation Framework. To support flexible feature iteration, we build the SFT pipeline on top of *Prompt Generation* (PG) [13], which offers a unified prompt template with configurable feature slots. This enables rapid feature experiments while ensuring training-serving consistency.

Data Construction. To construct data with QP-SID, the third-level token is jointly determined by the query q and the cluster of item i . Given a query q , we extract its key terms and match them against the representative terms $\mathcal{T}_{\text{cluster}}$ of item i 's cluster. When a match exists, we select the most representative matched term:

$$t^* = \arg \max_{t \in \mathcal{T}_{\text{cluster}} \cap \text{terms}(q)} \text{rel}(t, C) \quad (11)$$

where $\text{rel}(t, C)$ is the co-occurrence click count between term t and cluster C . The third-level token is then assigned as:

$$c_3(i, q) = \begin{cases} \pi_{t^*}(i) & \text{if } \mathcal{T}_{\text{cluster}} \cap \text{terms}(q) \neq \emptyset \\ \pi_0(i) & \text{otherwise} \end{cases} \quad (12)$$

making the third-level token a dynamic function of both the item and the query: an item is assigned an earlier index within its cluster when the query closely aligns with its dominant usage context, producing query-aware training labels that reflect fine-grained item-query relevance.

Table 2: Performance comparison of the proposed methods on the Taobao search dataset. Within each category, methods are mutually exclusive and [†] marks the configurations adopted in TSGR. Best results are in bold.

Category	Method	HR@20	HR@100	HR@500	HR@1000	HR@5000
SID	FORGE	0.4328	0.5863	0.7207	0.7735	0.8604
	QP-SID [†]	0.4635	0.6235	0.7669	0.8177	0.8961
Pre-processing	Pre-SFT [†]	0.4694	0.6340	0.7740	0.8222	0.8987
Post-processing	Pre-Ranking Model	0.4248	0.6264	0.8112	0.8552	0.8771
	RL	0.4659	0.6420	0.7857	0.8277	0.9027
	VRM (TSGR)[†]	0.4586	0.6677	0.8250	0.8651	0.8953

Session-wise Weighted Multi-Positive Training. For each search session, items from four engagement levels (pay, click, pv, and unpv) are packed into a single training sample, with interacted items (pay, click, pv) serving as multiple labels. Attention masks are applied between labels to prevent cross-label interference. Each label is assigned a priority weight satisfying $w_{\text{pay}} > w_{\text{clk}} > w_{\text{pv}}$. The generation loss is a weighted cross-entropy over all valid labels:

$$\mathcal{L}_{\text{gen}} = \sum_{k \in \mathcal{Y}} w_k \cdot \text{CE}(\mathbf{o}_k, t_k) \quad (13)$$

where $\mathcal{Y}$ is the set of interacted items in the session, $\mathbf{o}_k$ are the logits at the decoding positions of item k 's SID tokens, t_k is the ground-truth SID token, and w_k is the corresponding weight.

The VRM is trained jointly in this stage. Its ranking loss consists of three binary cross-entropy terms over the predicted scores for PV, CTR, and CVR, defined on the same $\mathcal{Y}$:

$$\mathcal{L}_{\text{rank}} = \sum_{k \in \mathcal{Y}} \sum_{s \in \{\text{pv}, \text{ctr}, \text{cvr}\}} \text{BCE}(\hat{y}_k^s, y_k^s) \quad (14)$$

where $\hat{y}_k^t$ is the predicted score and y_k^t is the corresponding label for task t . The overall objective combines the two terms:

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda \cdot \mathcal{L}_{\text{rank}} \quad (15)$$

Vanilla SFT aligns the model with observed user behavior, but falls short of directly optimizing downstream user satisfaction signals. To push performance further, we build a complete RL pipeline based on GRPO [18], using the online ranking model as the reward source to provide direct business-objective feedback. However, compared to introducing item side-info via the ranking module (Section 3.2), the RL-based approach makes limited additional gains. We therefore adopt the ranking module as our primary solution, with full RL experimental details provided in Appendix A.6.

4 Experiments

To more thoroughly assess the effectiveness of TSGR, in this section we conduct comprehensive offline and online A/B evaluations. Moreover, extensive ablation experiments are performed to prove the feasibility of each module.

4.1 Experimental Setup

Dataset and Baseline. We collect over 200 million real user interactions from Taobao's e-commerce search platform over two weeks as training data, with the test set sampled from the following

day. To assess generalizability of the VRM, we additionally conduct ablation experiments on it using the TencentGR [14] dataset in Appendix A.7. We compare against two baselines: **FORGE** [4], which follows its approach by randomly initializing the last-level token as the SID baseline, where each corresponds to up to 5 items; and **Pre-Ranking Model**, a conventional pre-ranking model applied after retrieval.

Evaluation Metrics. To verify retrieval performance, we adopt HitRate@K (HR@K) as evaluation metric. Following standard practice in industrial generative retrieval [1, 35], we report HR@K under the same beam size with **HR@1000** as the primary metric, as the goal of retrieval is to ensure that relevant items are surfaced within the candidate pool passed to downstream rankers.

Implementation Details. We adopt Qwen2.5-0.5B [26] as the generative backbone, with a three-level SID codebook (32768×8192×8192), beam sizes of [400, 400, 10000], and 4 parallel third-level orderings per item from QP-SID. SFT is trained with AdamW ($\beta_1=0.9$, $\beta_2=0.999$), a cosine learning rate of 1×10^{-4} , and a batch size of 256, with $\lambda=1$ and value score weights ($w_{\text{pv}}, w_{\text{clk}}, w_{\text{pay}}$) = (1, 2, 3). More details can be found in Appendix A.3.

4.2 Overall Comparison

Table 2 reports HR@K across all compared methods. We organize experiments around three dimensions: SID construction, pre-processing, and post-processing.

QP-SID outperforms FORGE across all metrics with +4.42% in HR@1000 by incorporating item value signals and query-conditioned orderings into the efficiency codebook. Within each cluster, higher-value items are assigned earlier token indices, which aggregate more items across clusters and thus receive more training signal. Beyond global value ordering, query-conditioned ordering further promotes query-relevant items to earlier indices, helping the model surface relevant candidates more effectively. Together, these two mechanisms ensure that the model prioritizes both globally high-value and query-specific relevant items.

The Pre-SFT multi-task pretraining stage further raises performance, demonstrating that this stage contributes substantial gains.

As a baseline for the post-processing stage, the conventional pre-ranking model delivers a notable improvement in HR@500 and HR@1000 (+3.30%) but incurs a performance loss in others. RL training, in contrast, achieves consistent improvements in model

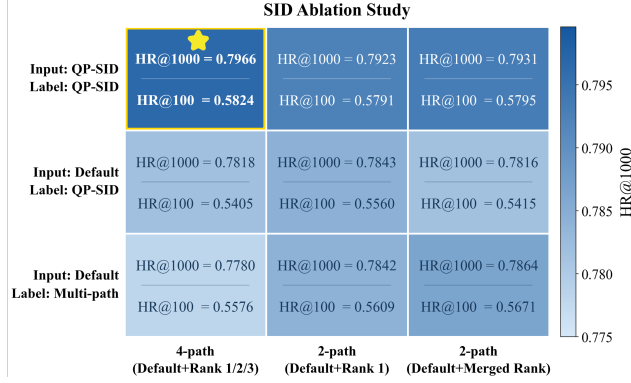


Figure 3: Comparison of different QP-SID configurations and training strategies. HR@1000 and HR@100 are reported for each combination of input/label configuration (rows) and efficiency ID strategy (columns).

capability across nearly all metrics (+0.55% in HR@1000). However, the magnitude of this improvement remains marginal, so RL has not been adopted in the actual deployment. The VRM achieves stronger performance instead by fusing backbone-derived user representations with item side-info under joint optimization with the generation objective, thereby avoiding cross-stage objective misalignment while introducing no additional serving latency. This module attains an improvement in HR@1000 comparable to that of the pre-ranking model (+3.73% vs. +3.30%), while outperforming it on nearly all others. Together, these results confirm that the unified approach of TSGR to retrieval and pre-ranking is both more effective and more deployment-efficient than the conventional cascaded design.

4.3 Ablation Study

To better examine the contribution of each design, we evaluate 1) the impact of SID construction strategies, 2) the effectiveness of the VRM, and 3) the key design choices within each training stage. For efficiency, part of the ablation studies are conducted on a sampled subset of the full training data.

4.3.1 SID Design. We progressively improve the SID design from FORGE to QP-SID. Re-ranking FORGE candidates with an efficiency score yields noticeable gains, but encoding efficiency directly into the SID via a dedicated codebook proves substantially more effective. We further reconstruct the semantic codebook with RQ-KMeans constrained by category and cluster priors (section 3.1), improving interpretability and extensibility. Consolidating the four-level codebook into three levels improves robustness against error accumulation, as a shallower SID is less susceptible to cascading prediction errors. Finally, QP-SID introduces query-aware parallel orderings to replace the single global efficiency ordering, selecting the best-matched ordering per query during serving.

Figure 3 investigates the impact of different QP-SID configurations and training strategies, revealing two key findings. First, consistency between input and output SID orderings is essential: using QP-SID orderings as training labels while retaining the default ordering as input weakens the model’s ability to copy the

Table 3: Hierarchical SID prediction accuracy of the top-100 output items under different label strategies.

Label Strategy	Cluster Hit Rate	Level-3 Hit Rate	Overall Hit Rate
QP-SID (Row 1)	78.71	72.83	57.32
Multi-path (Row 3)	77.20	71.90	55.50

Table 4: Distribution of QPSID categories across training and evaluation sets, as well as model output.

Category	Train Seq	Label	Eval Seq	Model Output
No Hit (Default)	55.37%	23.20%	61.42%	32.05%
Term Rank = 1	25.46%	54.54%	22.99%	44.91%
Term Rank = 2	11.63%	15.00%	9.71%	14.53%
Term Rank = 3	7.54%	7.26%	5.88%	8.51%

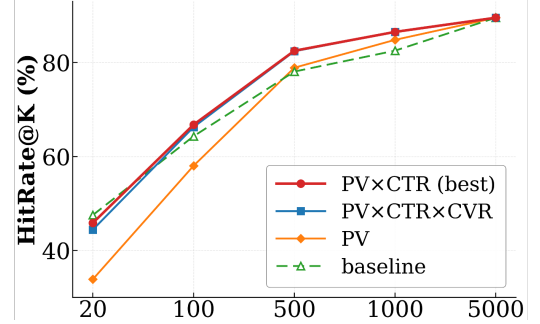


Figure 4: Ablation study on the effect of different score combinations in the VRM.

input SID sequence, causing a substantial HR@100 drop. Second, introducing auxiliary query-matched labels renders the third SID level harder to learn, amplifying its gradient and indirectly undermining the accuracy of the first two levels, as further evidenced by the hierarchical prediction accuracy in Table 3. Moreover, we observe that 99.8% of model outputs collapse to the Default path, indicating that the model fails to leverage the diversity of QP-SID orderings. These observations motivate our final QP-SID design.

Table 4 further validates it from a distributional perspective. The majority of items in behavior sequences fall into the Default category, indicating that most historically clicked items are not closely related to the current query. In contrast, up to 76.8% of training labels are query-relevant (Term Rank = 1, 2, or 3), which is expected since users click with the current query intent in mind. The model output distribution further confirms that the model has successfully learned to retrieve from multiple ranking channels rather than relying solely on the Default channel.

4.3.2 Value-aware Ranking Module. All experiments are conducted on a candidate pool of 5000 items. For the ranking score (Figure 4), we ablate different score combinations used for re-ranking at inference time. All combinations achieve consistent gains over the baseline on HR@1000, and PV × CTR attains the best overall performance. Given that PV × CTR directly corresponds to the HitRate

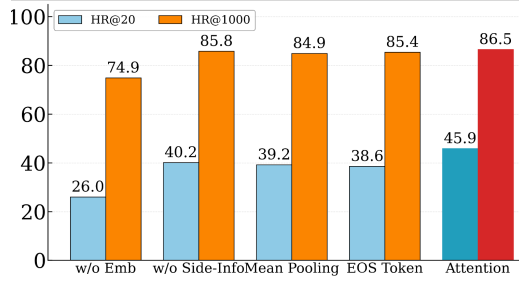


Figure 5: Ablation study on the effect of different architecture variants in the VRM.

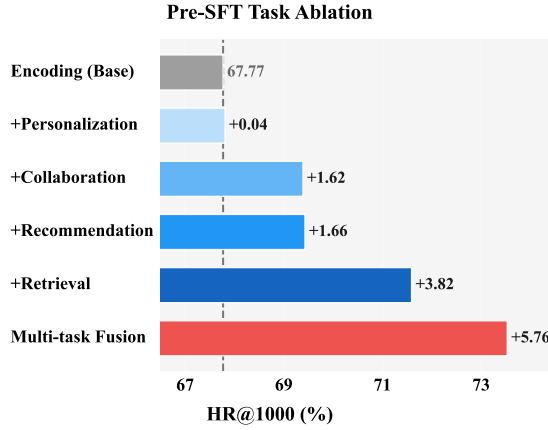


Figure 6: Ablation study on Pre-SFT task design. Each variant adds a single task on top of the Encoding base, while Multi-task Fusion trains all tasks jointly.

objective, this result is expected. Motivated by this observation, we further examine whether removing the CVR loss from training affects performance, and find that it has negligible impact in our setting. For the module architecture (Figure 5), removing item embeddings entirely causes severe degradation in both HR@20 and HR@1000, confirming that item representations are indispensable. This is because the SID construction process inevitably discards a substantial amount of item information, making item embeddings a critical complement for recovering the lost semantic detail. Excluding item side-info leads to a substantial drop in HR@20, with only moderate impact on HR@1000. Notably, both replacing the cross-attention module with mean pooling and substituting it with the EOS token representation directly passed into the MLP make a considerable decline in HR@20 while incurring only a marginal drop in HR@1000. This pattern suggests that these two designs can serve as viable lightweight alternatives in latency-sensitive scenarios. More comprehensive ablation results are provided in Appendix A.8.

4.3.3 Training Strategy. Figure 6 examines the contribution of each Pre-SFT task. Every task yields a positive gain, with the Retrieval task contributing the most due to its closest alignment with

Table 5: Ablation study on SFT training objectives.

Method	HR@20	HR@100	HR@500	HR@1k	HR@5k
Pointwise	0.3851	0.5653	0.7188	0.7714	0.8577
Listwise	0.4042	0.5890	0.7405	0.7922	0.8718

the downstream objective. The Personalization task contributes the least, as its title-generation format introduces an output-space gap relative to the SID target. Jointly training on all tasks achieves +5.76%, surpassing any individual task and confirming that the tasks activate complementary capability dimensions.

As shown in Table 5, the proposed session-wise weighted multi-positive objective consistently outperforms the pointwise baseline, confirming that explicitly modeling the relative priority among multiple positives leads to better sequence generation quality.

4.4 Online A/B Test

TSGR is deployed as an additional channel alongside existing retrieval sources, with its outputs passed directly to the ranking stage, bypassing the conventional pre-ranking stage. The system is served on NVIDIA H20 GPUs with a latency budget of 80ms. Given a user query and context features, the model performs constrained beam search. The first two SID levels are decoded under a valid-prefix constraint enforced by a pre-built prefix tree, which prunes invalid token sequences, while the third level uses standard beam search, yielding a set of complete SID sequences. As each candidate is generated, the VRM computes a value score. The candidates are then re-ranked and returned as the final output.

We deploy TSGR in a production A/B test on the Taobao search platform, serving 1% of live traffic over 38 consecutive days. In the treatment group, TSGR replaces the conventional retrieval and pre-ranking stages with a single model, directly forwarding its top-1,500 ranked candidates to the ranking stage. Compared with the production baseline, TSGR achieves gains of **+0.43%** in IPV, **+1.12%** in Transaction Count, and **+1.64%** in GMV. These improvements are statistically significant and, at the scale of Taobao Search, translate into substantial business value.

5 Conclusion

We present TSGR, a unified generative retrieval framework that addresses the value-insensitivity of existing GR systems in industrial e-commerce search. Our core insight is to embed value awareness into both item representation and model output: QP-SID encodes item efficiency directly into the semantic identifier, while a jointly optimized Value-aware Ranking Module enables a single model to serve as both retriever and pre-ranker, eliminating the objective misalignment of a separate pre-ranking stage. To further align the model with business objectives, we adopt a training pipeline of Pre-SFT, weighted multi-positive SFT, and explore the role of RL. Extensive offline experiments on Taobao search and online A/B tests demonstrate consistent and significant gains over strong baselines, and TSGR has been successfully deployed in production, serving real traffic at scale.

References

- [1] Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, et al. 2025. Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv preprint arXiv:2509.03236* (2025).
- [2] Ben Chen, Siyuan Wang, Yufei Ma, Zihan Liang, Xuxin Zhang, Yue Lv, Ying Yang, Huangyu Dai, Lingtao Mao, Tong Zhao, et al. 2026. OneSearch-V2: The Latent Reasoning Enhanced Self-distillation Generative Search Framework. *arXiv preprint arXiv:2603.24422* (2026).
- [3] Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740* (2024).
- [4] Kairui Fu, Tao Zhang, Shuwen Xiao, Ziyang Wang, Xinming Zhang, Chenchi Zhang, Yuliang Yan, Junjun Zheng, Yu Li, Zhihong Chen, et al. 2025. Forge: Forming semantic identifiers for generative retrieval in industrial datasets. *arXiv preprint arXiv:2509.20904* (2025).
- [5] Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, et al. 2025. Mtgr: Industrial-scale generative recommendation framework in meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 5731–5738.
- [6] Xuegang Hao, Ming Zhang, Alex Li, Xiangyu Qian, Zhi Ma, Yanlong Zang, Shijie Yang, Zhongxuan Han, Xiaolong Ma, Jinguang Liu, et al. 2025. OxygenREC: An Instruction-Following Generative Framework for E-commerce Recommendation. *arXiv preprint arXiv:2512.22386* (2025).
- [7] Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. 2025. Generating long semantic ids in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 956–966.
- [8] Yanhua Huang, Yuqi Chen, Xiong Cao, Rui Yang, Mingliang Qi, Yinghao Zhu, Qingchang Han, Yaowei Liu, Zhaoyu Liu, Xuefeng Yao, et al. 2025. Towards large-scale generative ranking. *arXiv preprint arXiv:2505.04180* (2025).
- [9] Jie Jiang, Xinxun Zhang, Enming Zhang, Yuling Xiong, Jun Zhang, Jingwen Wang, Huan Yu, Yuxiang Wang, Hao Wang, Xiao Yan, et al. 2026. End-to-End Semantic ID Generation for Generative Advertisement Recommendation. *arXiv preprint arXiv:2602.10445* (2026).
- [10] Zhijie Lin, Zhuofeng Li, Chenglei Dai, Wentian Bao, Shuai Lin, Enyun Yu, Haoxiang Zhang, and Liang Zhao. 2025. GRerF: A Unified Generative Framework for Efficient Reranking via Ordered Multi-token Prediction. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 5879–5887.
- [11] Gui Ling, Zhihong Chen, Yu Li, Tong Xiong, Kunhai Lin, Kaixuan Zhang, Yuliang Yan, Dan Ou, Haihong Tang, and Bo Zheng. 2026. Beyond Semantic IDs: Encoding Business-Value Ranking into Document Identifiers for Generative Retrieval. *arXiv:2607.11392 [cs.IR]* <https://arxiv.org/abs/2607.11392>
- [12] Xiaoyu Liu, Fuwei Zhang, Yiqing Wu, Xinyu Jia, Zenghua Xia, Fuzhen Zhuang, Zhao Zhang, Fei Jiang, and Wei Lin. 2026. CAT-ID2: Category-Tree Integrated Document Identifier Learning for Generative Retrieval In E-commerce. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*. 426–435.
- [13] Dan Ou, Gui Ling, Hao Wan, Hongbin Zhou, Jialiang Cheng, Jiangnan Pang, Silu Zhou, Wei Shi, Weichen Ye, Wenming Zhang, Yang Wang, Yu Li, Yuliang Yan, Zhan Fa, Zhihong Chen, Zongyuan Wu, Bo Zheng, Changfa Wu, Dunxian Huang, Haihong Tang, Jinlong Guo, Kaixuan Zhang, Kun Ma, Lin Qu, Longbo Zhong, Tao Lan, Tong Xiong, and Zhibo Wu. 2026. Prompt Generation Technical Report. *arXiv:2607.11326 [cs.IR]* <https://arxiv.org/abs/2607.11326>
- [14] Junwei Pan, Wei Xue, Chao Zhou, Xing Zhou, Lunan Fan, Yanbo Wang, Haoran Xin, Zhiyu Hu, Yaozhong Wang, Fengye Xu, et al. 2026. Tencent Advertising Algorithm Challenge 2025: All-Modality Generative Recommendation. *arXiv preprint arXiv:2604.04976* (2026).
- [15] Changhua Pei, Xinru Yang, Qing Cui, Xiao Lin, Fei Sun, Peng Jiang, Wenwu Ou, and Yongfeng Zhang. 2019. Value-aware recommendation based on reinforcement profit maximization. In *The World Wide Web Conference*. 3123–3129.
- [16] Aleksandr V Petrov and Craig Macdonald. 2024. RecJPQ: training large-catalogue sequential recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 538–547.
- [17] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.
- [18] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).
- [19] Chaotian Song, Jingyao Zhang, Chenghao Chen, Zisen Sang, Dehai Zhao, Guodong Cao, Boxi Wu, Deng Cai, and Jia Jia. 2026. DeGR: Dense-supervised Generative Reranking for Recommendation. *arXiv preprint arXiv:2605.25749* (2026).
- [20] Yanglong Song, Zihao Yang, Shuo Meng, Rujun Guo, Jin Zhang, Bin Wang, Shaoyu Liu, Xiaozhao Wang, and Guanjun Jiang. 2026. From Item-Only to Query-Item: Query-Conditioned Generative Search with QGS in Quark. *arXiv preprint arXiv:2605.25514* (2026).
- [21] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Idgenrec: Llm-recsys alignment with textual id learning. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 355–364.
- [22] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in neural information processing systems* 35 (2022), 21831–21843.
- [23] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2400–2409.
- [24] Yifan Wang, Yixuan Wang, YiDan Liang, Qiang Liu, and Fei Xiao. 2026. Towards Sustainable Growth: A Multi-Value-Aware Retrieval Framework for E-Commerce Search. *arXiv preprint arXiv:2605.17994* (2026).
- [25] Yanjing Wu, Yinfu Feng, Jian Wang, Wenji Zhou, Yunan Ye, Rong Xiao, and Jun Xiao. 2024. Hi-gen: Generative retrieval for large-scale personalized e-commerce search. *arXiv preprint arXiv:2404.15675* (2024).
- [26] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 Technical Report. *arXiv e-prints* (2024), arXiv:2412.
- [27] Haiyang Yang, Qinye Xie, Qingheng Zhang, Chen Li Yu, Huike Zou, Chengbao Lian, Shuguang Han, Fei Huang, Jufeng Chen, and Bo Zheng. 2025. GSID: Generative Semantic Indexing for E-Commerce Product Understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 1113–1121.
- [28] Jiaqi Zhai et al. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. (2024).
- [29] Jun Zhang, Yi Li, Yue Liu, Changping Wang, Yuan Wang, Yuling Xiong, Xun Liu, Haiyang Wu, Qian Li, Enming Zhang, et al. 2025. GPR: Towards a Generative Pre-trained One-Model Paradigm for Large-Scale Advertising Recommendation. *arXiv preprint arXiv:2511.10138* (2025).
- [30] Kaike Zhang, Xiaobei Wang, Shuchang Liu, Hailan Yang, Xiang Li, Lantao Hu, Han Li, Qi Cao, Fei Sun, and Kun Gai. 2025. Goalrank: Group-relative optimization for a large ranking model. *arXiv preprint arXiv:2509.22046* (2025).
- [31] Xinxun Zhang, Yuling Xiong, Jiale Zhou, Zhengkai Guo, Zhennan Pang, Junbang Huo, Jingwen Wang, Xuyang Sun, Enming Zhang, Jianguang Jin, et al. 2026. Unified Value Alignment for Generative Recommendation in Industrial Advertising. *arXiv preprint arXiv:2605.05803* (2026).
- [32] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1435–1448.
- [33] Zuowu Zheng, Ze Wang, Fan Yang, Jiangke Fan, Teng Zhang, Yongkang Wang, and Xingxing Wang. 2025. Ega-v2: An end-to-end generative framework for industrial advertising. *arXiv preprint arXiv:2505.17549* (2025).
- [34] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. 2025. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* (2025).
- [35] Jianbo Zhu, Xing Fang, Jing Wang, Mingmin Jin, Bokang Wang, Guangxin Song, Zhenyu Xie, and Junjie Bai. 2026. Efficient Generative Retrieval for E-commerce Search with Semantic Cluster IDs and Expert-Guided RL. *arXiv preprint arXiv:2605.14434* (2026).

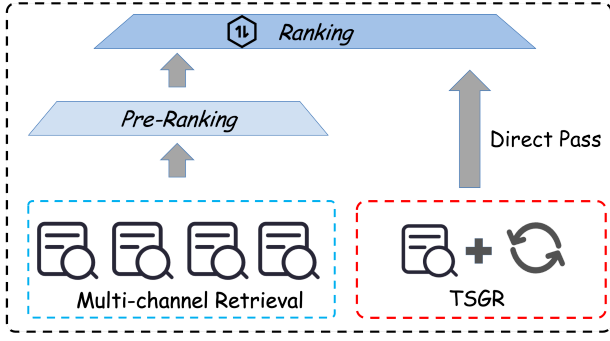


Figure 7: Online deployment architecture of TSGR. Items generated by TSGR bypass pre-ranking and enter the ranking stage directly.

A Appendix

A.1 The Use of Large Language Models (LLMs)

In this paper, we use the publicly available LLM Qwen as the backbone model of our experiments. As for writing, we use generative AI tools such as Grammarly solely to improve the clarity, coherence, and overall quality of the written text. All figures, preliminary drafts, and content are written independently by the authors without reliance on LLMs.

A.2 Online Deployment

As illustrated in Figure 7, TSGR is deployed as an independent retrieval channel alongside multi-channel retrieval in the Taobao Search pipeline, with the key distinction that its output bypasses pre-ranking and enters the ranking stage directly, given that the generated items are already ordered by value scores that reflect both relevance and commercial quality.

A.3 Implementation Details

The batch sizes for SFT and GRPO are 256 and 16 (2 per device $\times$ 8 gradient accumulation steps), respectively. SFT employs AdamW ($\beta_1=0.9$, $\beta_2=0.999$) with a cosine learning rate of 1×10^{-4} and 10 warmup steps, whereas GRPO employs a linear learning rate of 4×10^{-7} with 20 warmup steps over 2 epochs on 100K queries. Inference applies dynamic beam search with sizes [400, 400, 10000] across the three SID levels, and λ is set to 1. For GRPO, 32 candidates are sampled per query at a temperature of 1.0 under constrained decoding; the top-10 candidates by ranking CTR $\times$ CVR score receive a reward of +1 and the bottom-10 receive a reward of -1, with 16 offline preference pairs appended per prompt. The KL coefficient is set to 0.5 and is scaled to $0.6 \times$ when the KL and policy gradient directions conflict at the sample level. Entropy regularization (0.005) is applied to Levels 1–2 and disabled when the overlap exceeds 20%.

A.4 QP-SID Case Study

To intuitively illustrate how QP-SID organizes items, Figure 8 presents a case study on two representative clusters: phone cases (Cluster (140, 3504)) and dining tables (Cluster (10278, 7631)). For each cluster, we compare the *default* value ordering with the *query-conditioned* ordering under a representative term, namely “磁吸”

(magnetic) for phone cases and “实木” (solid wood) for dining tables. Items within each row are sorted in descending order of their value scores. Note that the default and query-conditioned scores are computed under different criteria and are therefore not directly comparable; only the relative ordering within each row is meaningful.

The comparison reveals two complementary behaviors of QP-SID. The default ordering reflects the overall commercial value of items within a cluster, ranking globally popular and high-value items first. In contrast, the query-conditioned ordering re-prioritizes items according to query intent: under the term “磁吸” or “实木”, items whose titles explicitly match the term (highlighted in red) are promoted to the top positions, even if they rank lower in the default ordering. This demonstrates that QP-SID maintains multiple query-aware value orderings for the same cluster, enabling the model to select the ordering that best matches the dominant intent of a given query and thus place intent-aligned high-value items in more favorable token positions.

Figure 9 further illustrates how QP-SID selects among these orderings during serving, where each ordering is indexed by a term derived from term-item click statistics (Section 3.1). We consider two representative cases. In the single-match case (Figure 9, left), each query matches exactly one term, which directly determines the Level-3 SID assigned to the target item. For example, the queries “分体独立电话机” and “座机电话家用无绳” match the terms “电话机” and “座机”, respectively, showing that semantically distinct queries pointing to the same item can be routed through different terms, each reflecting the specific intent of the query. In the multi-match case (Figure 9, right), a query may match several terms simultaneously, and QP-SID selects the top-ranked term to govern the value ordering while discarding the rest. For instance, the query “百乐capless钢笔十代” matches three terms (“capless”, “百乐”, and “钢笔”), and the top-ranked term “capless” (solid arrow) is selected while the others are discarded (dashed arrows). Taken together, these cases show that QP-SID flexibly adapts its value ordering to query intent, ensuring that the most intent-relevant ordering guides SID generation during serving.

A.5 Prompt Templates for the Pipeline

Table 6 presents the templates for the five Pre-SFT task types, and Table 7 presents the unified template shared by the SFT and RL stages. Each template specifies the natural-language instruction together with placeholders, denoted by $\{\cdot\}$, that are filled with the corresponding item attributes (e.g., title, semantic ID) or user attributes (e.g., gender, age, purchasing power, historical interaction sequence) at training time. All prompts are originally written in Chinese, consistent with the Taobao Search scenario, and are presented here in their original form to faithfully reflect the actual model inputs.

A.6 Details of RL

Reward Design. The reward steers generation toward candidates that are both relevant and high in predicted business value, defined as the product of CTR and CVR from the ranking model:

$$r(s) = \text{CTR}(q, i_s) \times \text{CVR}(q, i_s). \quad (16)$$



Figure 8: Case study of QP-SID on two clusters: phone cases (Cluster (140, 3504)) and dining tables (Cluster (10278, 7631)). For each cluster, we show the default value ordering (red) and the query-conditioned ordering under a specific term (blue), with matched terms highlighted in red. Scores beside each item denote the corresponding value scores.



Figure 9: Case study of term selection in QP-SID during serving. Left: different queries each match a single term, which determines the corresponding Level-3 SID. Right: when a query matches multiple terms, QP-SID selects the top-ranked term (solid arrow) and discards the others (dashed arrows). Matched terms are highlighted in red.

Offline DPO Augmentation. Standard GRPO relies on on-policy rollouts to form group-relative rewards, but these rollouts are typically concentrated on head candidates. As a result, reward normalization systematically suppresses lower-reward samples, even though some of them may still fall within the top-1000 region of interest. While this bias may benefit head-heavy metrics, it is sub-optimal for our target retrieval objective. To mitigate this issue, we augment each prompt with offline preference pairs constructed from ranking CTR×CVR scores: chosen samples are drawn from the ranking top- B candidates and ground-truth purchase labels, while rejected samples are sampled from candidates around rank 1000 to reduce the risk of false negatives. Furthermore, after mixing the on-policy rollouts with the offline preference samples, we retain only the top- k and bottom- k samples in the merged set by reward

($k = 10$), and mask out the rewards of all remaining samples. This design concentrates optimization on high-confidence preference signals while preventing GRPO’s group-relative normalization from mistakenly suppressing ambiguous candidates that may still belong to the top-1000 set.

Dynamic Hierarchical Entropy Regularization. To encourage exploration of diverse SID paths, we apply entropy regularization with a dynamic, layer-wise strategy. For each prompt, we measure the match rate m between the rollout SIDs and the ground-truth SIDs at Levels 1–2, and gate the regularization with a binary coefficient:

$$\beta(m) = \begin{cases} \beta_0 & \text{if } m < \tau_{\text{low}}, \\ 0 & \text{if } m > \tau_{\text{high}}, \end{cases} \quad (17)$$

Table 6: Prompt templates for the five Pre-SFT task types.

Task Type	Task Definition
Encoding	商品的标题为: $\${title}$, 请预测它的语义id
Personalization	你是一个电商搜索引擎, 你的任务是根据用户的个人信息、当前的搜索词、用户历史交互的商品序列, 为用户检索下一个商品。用户的性别为: $\${sex}$, 年龄为: $\${age}$, 购买力级别为: $\${level}$ 。用户历史交互的商品序列为: $\${sids}$ 。你检索的商品标题是
Retrieval	你是一个电商搜索引擎, 你的任务是根据当前的搜索词, 为用户检索商品。搜索词为: $\${query}$ 。你检索的商品是
Collaboration	某商品的标题为: $\${trigger_title}$, 它的语义id为: $\${trigger_sid}$, 购买过它的用户通常还购买过相似商品, 标题为: $\${target_title}$, 请预测这一相似商品的语义id
Recommendation	你是一个电商推荐引擎, 你的任务是根据用户的个人信息、用户历史交互的商品序列, 为用户推荐下一个商品。用户的性别为: $\${sex}$, 年龄为: $\${age}$, 购买力级别为: $\${level}$ 。用户历史交互的商品序列为: $\${sids}$ 。你推荐的商品是

Table 7: SFT and RL prompt template for personalized retrieval.

Role	Content
System	请根据用户信息、搜索词、用户历史交互商品序列, 为用户检索下一个商品。
User	用户性别为: $\${gender}$, 年龄为: $\${age}$, 购买力级别为: $\${level}$ 。搜索词为: $\${query}$ 。历史交互商品序列为: $\${opt_seq}$ 。你检索的商品是

Table 8: Ablation study of RL strategies. Each row cumulatively adds one component upon the previous configuration.

Method	HR@20	HR@100	HR@500	HR@1k	HR@5k
Baseline	0.4965	0.6580	0.7746	0.8111	0.8747
+ Basic RL	0.4985	0.6584	0.7761	0.8113	0.8737
+ DPO Augmentation	0.4975	0.6658	0.7803	0.8152	0.8772
+ Dynamic Entropy Reg	0.4989	0.6674	0.7808	0.8163	0.8788

so that exploration is restricted to prompts that have not yet converged. We apply it only to Levels 1–2, since entropy at Level 3 would perturb the fine-grained item distribution calibrated by SFT. The resulting term is added to the GRPO objective:

$$\mathcal{L}_{RL} = \mathcal{L}_{GRPO} - \beta(m) \sum_{l \in \{1,2\}} \mathcal{H}(\pi_{\theta}^{(l)}), \quad (18)$$

where $\mathcal{H}(\pi_{\theta}^{(l)})$ denotes the entropy of the policy distribution at Level l .

Table 8 shows that each RL component contributes incrementally. Basic RL alone yields only marginal gains (+0.02%). DPO Augmentation brings a more notable improvement (+0.39%), indicating that preference-based supervision helps the model better distinguish

Table 9: Ablation experiment on the TencentGR-10M dataset.

Dataset	Method	HR@500
TencentGR-10M	Base	0.0786
	Value-aware Ranking Module	0.0794

between positive and negative candidates. Dynamic Entropy Regularization further improves performance (+0.11%), confirming that maintaining output diversity during RL training is beneficial. The modest cumulative gain of +0.52% explains why RL has not been adopted in actual deployment.

A.7 TencentGR Dataset

The TencentGR dataset [14] poses several practical challenges that require dedicated preprocessing before it can be used to evaluate our framework. Specifically, the dataset provides no query field, no pre-built semantic IDs, fully anonymized user and item attributes, and no public test set with ground-truth labels. We describe below how each challenge is addressed.

SID Construction. Since the dataset does not provide semantic IDs, we construct them directly from the 1024-dimensional multimodal

Table 10: Ablation study of the Value-aware Ranking Module.

Model Design	Ranking Strategy	HR@20	HR@100	HR@500	HR@1000	HR@5000
Attention	Base	0.4754	0.6431	0.7807	0.8252	0.8953
	PV	0.3387	0.5803	0.7889	0.8482	0.8953
	CTR	0.3744	0.5401	0.7088	0.7824	0.8953
	CVR	0.2149	0.3097	0.4717	0.5802	0.8953
	PV×CTR	0.4586	0.6677	0.8250	0.8651	0.8953
	PV×CTR×CVR	0.4443	0.6626	0.8239	0.8650	0.8953
Mean Pooling	Base	0.4774	0.6492	0.7822	0.8269	0.8985
	PV	0.2726	0.5273	0.7599	0.8303	0.8985
	CTR	0.3148	0.4995	0.7020	0.7865	0.8985
	CVR	0.1863	0.3002	0.4825	0.6008	0.8985
	PV×CTR	0.3920	0.6161	0.8003	0.8488	0.8985
	PV×CTR×CVR	0.3849	0.6146	0.7999	0.8503	0.8985
EOS Token	Base	0.4763	0.6493	0.7840	0.8272	0.8948
	PV	0.3587	0.5975	0.7942	0.8483	0.8948
	CTR	0.3455	0.5278	0.7227	0.7977	0.8948
	CVR	0.2418	0.3662	0.5448	0.6514	0.8948
	PV×CTR	0.3862	0.6058	0.8016	0.8539	0.8948
	PV×CTR×CVR	0.3727	0.5916	0.7953	0.8512	0.8948
No Embedding	Base	0.4775	0.6432	0.7790	0.8233	0.8954
	PV	0.2236	0.4016	0.6336	0.7390	0.8954
	CTR	0.2123	0.3342	0.5092	0.6053	0.8954
	CVR	0.1548	0.2355	0.3656	0.4613	0.8954
	PV×CTR	0.2600	0.4356	0.6448	0.7487	0.8954
	PV×CTR×CVR	0.2621	0.4384	0.6483	0.7502	0.8954
No Side-Info	Base	0.4763	0.6469	0.7803	0.8269	0.8963
	PV	0.3868	0.6097	0.7922	0.8429	0.8963
	CTR	0.3580	0.5382	0.7194	0.7913	0.8963
	CVR	0.1645	0.2404	0.4096	0.5432	0.8963
	PV×CTR	0.4116	0.6412	0.8139	0.8580	0.8963
	PV×CTR×CVR	0.4013	0.6341	0.8125	0.8570	0.8963

item embeddings. We apply hierarchical clustering to produce a two-level SID prefix, and determine the third level by ranking items within each cluster in descending order of their transaction, click, and impression counts. Each item is thus mapped to a three-level SID.

Item-Side Information. As raw item content is not publicly available, item-side information is assembled from three sources: (1) the constructed SID; (2) anonymized item features concatenated with per-item impression, click, and transaction counts aggregated from the behavior sequences (missing values are filled with -1); and (3) the ℓ_2 -normalized 1024-dimensional multimodal embedding.

Behavior Sequence Preprocessing. The raw behavior sequences contain approximately 10.1M records, many of which consist entirely of impressions with no clicks or transactions and are therefore uninformative for training or evaluation. After filtering out these impression-only records, approximately 7.71M records are retained.

Each remaining record is then processed for multi-label training: the most recent impression, click, and transaction are each selected as a label; to prevent label leakage, only behavior that occurs before the earliest label timestamp is used as input; and the input is split into three separate sub-sequences corresponding to impressions, clicks, and transactions, respectively.

Prompt Construction. As no interpretable user attributes are available, all user features are included in the prompt in the form Feature N : value without semantic distinction. The prompt instructs the model to recommend the next item given the user features and the three historical sub-sequences encoded as SID strings. The output consists of up to three SID strings corresponding to the transaction, click, and impression labels in order of priority, with lower-priority labels used as fallbacks when higher-priority ones are absent. Each training sample additionally carries the multimodal

embedding and side information of the label items, a label priority indicator, and group identifiers used for multi-label alignment.

Train/Test Split. As no public test set is available, we construct a reproducible split by sorting all records by MD5(user_id) and designating the first 20,000 records as the test set, with the remainder used for training. This procedure ensures that users in the test set do not appear in the training set.

A.8 Other Experiments

We provide a comprehensive ablation study of the value-aware ranking module in Table 10. We evaluate different ranking strategies, including PV, CTR, CVR and their combinations, across multiple model variants. Results show that the Attention-based model with $PV \times CTR$ ranking achieves the best overall performance.