

WeGR: Unified Generative Retrieval for Industrial Search Systems in WeChat

Jiajun Han*, Guoxin Ma*, Felix[†], Liangsheng Xu[#], Ruiqi Yu[#], Yingdi Guo, Zongbing Wang, Xinhua Ye, Chenyan Hua, Xin Xie, Pengcheng Yang, Haijun Wu

Abstract

Generative retrieval (GR) offers a promising paradigm for industrial search by unifying candidate generation within an autoregressive framework. However, its effectiveness is limited by two key challenges. First, existing Semantic ID (SID) tokenizers struggle to incorporate collaborative signals while maintaining effective codebook utilization, limiting their ability to distinguish items and achieve high-quality retrieval. Second, existing reinforcement learning (RL) methods suffer from severe reward sparsity under exact SID matching, leaving many sampled trajectories without effective learning signals and hindering preference optimization. To address these challenges, we propose WeGR, a unified generative retrieval framework consisting of two complementary components. Specifically, the Collaborative and Collapse-free Tokenizer (CoCo-VAE) incorporates query-to-item and item-to-item collaborative signals while mitigating codebook collapse through codebook refresh and complementary regularization, enabling more informative and balanced SID representations. Building upon these SIDs, Dynamic Horizon Policy Optimization (DHPO) progressively expands both reward evaluation and policy optimization horizons, reducing learning difficulty while leveraging behavioral reward signals at different levels of the SID hierarchy, without relying on auxiliary reward models. Extensive offline experiments demonstrate the effectiveness of WeGR in improving retrieval and ranking performance. Moreover, a two-month online A/B test in WeChat E-commerce Search demonstrates relative improvements of 2.539% in orders, 1.508% in click-through rate, and 1.381% in conversion rate, while reducing online CPU and GPU usage by 25.5% and 26.1%, respectively. The system has been successfully deployed across multiple search scenarios in WeChat, serving billions of users, generating tens of millions of PVs daily.

Keywords

Search System, Generative Retrieval, Reinforcement Learning

1 Introduction

As illustrated in Figure 1, modern industrial search systems typically organize candidate generation, pre-ranking, and ranking as a multi-stage cascade. Such systems maintain multiple retrieval channels and pass their merged candidate set through downstream ranking stages, increasing both serving complexity and computational cost. Generative retrieval (GR) offers an alternative paradigm by representing each item as a sequence of discrete tokens, commonly referred to as a Semantic ID (SID), and directly generating candidate SIDs conditioned on the query and user context [1, 25, 29, 33]. By replacing independent candidate scoring with autoregressive

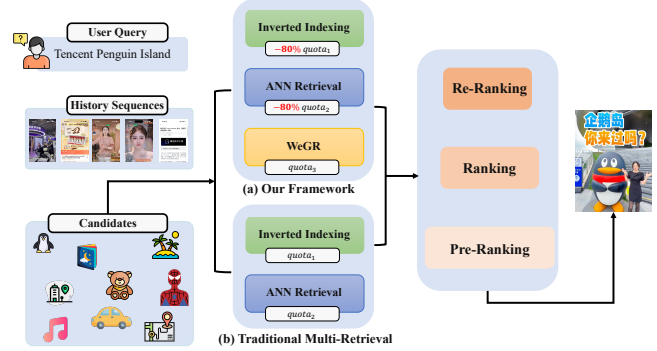


Figure 1: (a) Our proposed End-to-End generative retrieval framework, (b) the traditional multi-stage cascading architecture in E-commerce search.

generation, GR provides a unified interface for modeling relevance, semantics, and user preference. Realizing this potential in large-scale industrial search system, however, depends on learning behaviorally informative SIDs and aligning the policy model with observed user feedback in a robust and transferable manner.

The first challenge lies in constructing SIDs that preserve collaborative semantics while maintaining effective codebook utilization. Existing quantization methods mainly follow two paradigms. RQ-VAE-based methods support end-to-end learning and allow task-specific supervision to directly shape SID construction, but discrete assignment and successive residual decomposition can produce imbalanced codeword usage and codebook collapse [25, 31]. RQ-Kmeans-based methods provide more explicit control over cluster assignments and can incorporate balancing strategies or alternative quantizers [2, 9]; however, collaborative and business-specific signals are usually encoded only in upstream item representations and may be attenuated or distorted during multi-level residual quantization. Consequently, existing paradigms offer complementary strengths but do not fully reconcile collaborative-information preservation with stable codebook utilization, limiting the representational capacity of the resulting SID space.

The second challenge lies in aligning the generative policy with user preferences. Supervised fine-tuning (SFT) maximizes the likelihood of observed positive SIDs but does not explicitly distinguish the relative value of generated candidates. On-policy reinforcement learning (RL) facilitates preference optimization by leveraging reward signals from user interactions [12, 37]. Group Relative Policy Optimization (GRPO) is a widely used optimizer that learns from the relative advantages of multiple sampled trajectories. However, when using outcome-based rewards from clicks and orders, GRPO requires generated SIDs to exactly match positive items, resulting in severe reward sparsity and insufficient learning signals [35]. To

*These authors contributed equally as co-first authors.

[#]These authors contributed equally as co-second authors.

[†]Corresponding author.

alleviate reward sparsity, existing approaches either expand rollout budgets via guided decoding (e.g., beam search) [4] or introduce auxiliary rankers for denser rewards [3, 36]. Larger rollout budgets incur substantial computation, while beam search introduces a mismatch between selected trajectories and the policy’s sampling distribution. On the other hand, auxiliary rankers—trained strictly on candidates from upstream retrieval stages—provide unreliable scores for medium- and long-tail items explored by the generative policy. Furthermore, practical ranking scores often rely on hand-crafted heuristics tailored to specific scenarios, limiting cross-scenario generalizability and requiring costly adaptation. Thus, denser rewards do not necessarily yield better preference alignment and may instead inject misleading supervision.

To address these challenges, we present **WeGR**, a unified generative retrieval framework for industrial search systems. WeGR connects behavior-aware SID construction with reward-model-free preference optimization through two components. First, the **Collaborative and Collapse-free Tokenizer (CoCo-VAE)** incorporates query-to-item and item-to-item collaborative supervision into continuous representation learning. It further combines first-layer codebook refresh with residual-fidelity and collision regularization, mitigating persistent codeword underutilization while preserving semantic structure across quantization layers. Second, **Dynamic Horizon Policy Optimization (DHPO)** leverages the hierarchical structure of SIDs for coarse-to-fine GRPO optimization. By progressively expanding the horizons for both reward evaluation and policy optimization, DHPO allows sampled trajectories to receive behavioral rewards through prefix matching during early training, before gradually transitioning to exact item-level preference optimization. This enables more effective use of click- and order-based outcome rewards without auxiliary reward models. We further introduce Multi-Positive Supervised Fine-Tuning (MSFT) to learn from multiple positive items and provide stable learning signals when group-relative rewards become uninformative.

We evaluate WeGR in large-scale production search. In controlled online A/B tests, the V1.2 deployment removes 80% of candidates from conventional retrieval channels while achieving relative improvements of 2.539%, 1.508%, and 1.381% in orders, CTR, and CVR, respectively. This reduces the number of candidates entering pre-ranking by 38.10%, yielding CPU and GPU savings of 25.5% and 26.1%. WeGR also reduces average serving latency by 21.7 ms in WeChat Channels Search and 15.4 ms in WeChat Search. Further analyses examine performance across different user groups and query-frequency levels. To validate its cross-scenario applicability, we independently evaluate DHPO on WeChat e-commerce live-stream search, where it consistently outperforms SFT across all Click and Order metrics, achieving relative gains of 5.95%–7.07% in HR@6 and 6.12%–7.82% in MRR@6. These results demonstrate the effectiveness and efficiency of WeGR in industrial deployment, as well as the applicability of DHPO across different search scenarios. The main contributions of this work are summarized as follows:

- We introduce **CoCo-VAE**, a behavior-aware residual quantization framework that integrates collaborative supervision with codebook refresh, residual-fidelity preservation, and collision regularization to mitigate codebook underutilization without sacrificing SID semantics.

- We propose **DHPO**, a reward-model-free GRPO variant that progressively expands the horizons of reward evaluation and policy optimization along the SID hierarchy, alleviating outcome reward sparsity while maintaining generalizability across diverse search scenarios.
- We demonstrate the effectiveness of WeGR via production experiments and fine-grained analyses, showing improvements in business metrics with reduction in candidate volume, serving resources, and latency. We further validate DHPO in an independent live-stream search scenario.

2 Related Works

2.1 Generative Retrieval

Generative retrieval (GR) formulates retrieval as autoregressive generation, directly predicting target identifiers conditioned on queries or user histories [1, 14, 29, 33]. For item retrieval, each item is represented as a Semantic ID (SID), a sequence of discrete tokens encoding its semantic information. Such SIDs are constructed through two paradigms: RQ-VAE-based learnable quantization [17, 25] and RQ-Kmeans-based residual clustering [2, 9].

RQ-Kmeans-based methods. [20] construct SIDs by successively clustering item embeddings and residuals, with explicit assignment control. OneRec [9] uses balanced clustering, while OneSearch [2] combines selective cluster balancing with optimized product quantization of the remaining residual [13]. These designs separate quantization from behavior-aware representation learning. In OneSearch, collaborative signals enter upstream contrastive embeddings, but subsequent clustering does not directly optimize behavioral objectives. Preserving these signals in SIDs thus relies on quantization fidelity, not direct behavioral supervision of code assignment.

RQ-VAE-based methods. learn representations and codebooks via reconstruction and quantization, allowing task supervision. LC-Rec [38] combines learned indices with collaborative-semantic alignment tasks, while LETTER [32] directly regularizes the tokenizer via collaborative contrastive alignment and code-assignment diversity. ETEGRec [19] couples tokenization and recommendation via sequence–item and preference–semantic alignment. These approaches differ in whether alignment targets the recommender, the tokenizer, or their joint training, rather than merely in the information used. Yet task alignment cannot ensure effective codebook utilization: hard assignments can undertrain rare codewords, causing codebook collapse [30] [21]. DIGER [11] enables recommendation-guided differentiable assignment with Gumbel-noise exploration to reduce code concentration; R3-VAE [31] stabilizes quantization via reference-vector guidance and scoring-based assignment.

CoCo-VAE jointly addresses behavioral alignment and codebook underutilization within the RQ-VAE framework. It combines query-to-item and item-to-item supervision in continuous representations with codebook maintenance and regularization to preserve collaborative semantics, quantization fidelity, and codebook utilization.

2.2 Preference Alignment for Generative Retrieval

Preference alignment in GR aims to capture user preferences and downstream utility, such as relevance, engagement, and conversion, beyond generation likelihood. Existing approaches achieve this

through either inference-time guidance or training-time optimization. Given predicted semantic-code values, GPR's Value-Guided Trie-based Beam Search [34] allocates beam widths under trie constraints without updating the generative policy.

Training-based approaches [18] incorporate utility into supervised or preference objectives. Exponential reward-weighted SFT [7] reweights logged interactions by observed rewards, while VARG [8] combines value-aware supervision with local ordinal learning over value-ordered codes. DPO [24] learns relative candidate preferences; RankGR [12] extends comparisons to lists, and RAD-DPO [6] addresses shared-prefix conflicts and noisy implicit feedback.

Reinforcement learning aligns generation with retrieval utility through relevance-feedback training [41], PPO [26], or GRPO [27]. However, exact-match behavioral rewards require the complete SID of a clicked or ordered item, leaving partially correct prefixes unrewarded [35]. Limited rollouts can yield all-zero reward groups and no reward-driven policy gradient. With advantages centered within each group, increasing sample count alone cannot recover a learning signal if all sampled rewards remain identical.

Existing methods alleviate reward sparsity by modifying sampling or reward design [39]. Larger rollout budgets increase computational costs, while HCGRec [35] uses target-prefix hints to recover rewarding trajectories but introduces additional training complexity and a potential training-inference mismatch. VARG [8] provides denser feedback through relevance and ranker-derived rewards, yet its reliance on auxiliary scoring systems may limit cross-scenario generalizability.

Hierarchy-aware methods alleviate reward sparsity through finer-grained credit assignment. OneSearch-V2 [3] combines behavioral rewards with estimated relevance and CTR, using position-wise advantages and prefix-dependent gating. HRPO [15] estimates prefix utilities from user-group statistics to derive token-level rewards. However, both methods rely on additional utility estimation or reward design, potentially introducing noisy supervision and limiting generalizability across search scenarios.

DHPO alleviates reward sparsity through curriculum learning by progressively expanding reward evaluation and policy optimization horizons from prefix-level to item-level matching. This enables effective behavioral supervision while ensuring accurate credit assignment, without relying on auxiliary reward models.

3 Methodology

In this section, we present WeGR, a unified generative retrieval framework for e-commerce search. WeGR consists of two complementary components. First, CoCo-VAE (§ 3.1) incorporates query-to-item and item-to-item collaborative signals into SID construction while mitigating codebook collapse through complementary regularization. Building on these SIDs, DHPO (§ 3.2.1) progressively expands reward evaluation and policy optimization horizons, enabling coarse-to-fine preference learning under sparse outcome rewards without relying on auxiliary reward models. The overall framework is illustrated in Figure 2.

3.1 Collaborative and Collapse-free Tokenizer

To preserve collaborative information while maintaining effective codebook utilization, we propose CoCo-VAE, a collaborative-aware

residual quantization framework. Specifically, CoCo-VAE incorporates query-to-item (Q2I) and item-to-item (I2I) collaborative signals to guide representation learning, while combining Codebook Refresh with Fidelity Anti-collapse Regularization (FAR) to mitigate codebook collapse during hierarchical quantization.

3.1.1 Q2I And I2I Collaborative Regularization. We introduce Q2I and I2I collaborative signals to address the lack of co-occurrence information in multimodal embeddings [40]. Specifically, an auxiliary contrastive loss is employed to capture collaborative relationships and guide SID construction. Unlike existing methods that impose constraints directly within the discrete codebook space, which can cause codebook collapse by forcing semantically dissimilar embeddings to share the same SID, we apply such constraints exclusively to the continuous representations of the encoder with a detached projection head. This design decouples collaborative representation learning from codebook optimization, allowing behavioral signals to enrich item representations without disrupting codebook assignments, thereby preserving representation diversity, quantization fidelity, and codebook stability.

3.1.2 Codebook Refresh. In residual quantization, some codewords may become “dead” due to insufficient usage during training, limiting effective codebook capacity. Incorporating Q2I and I2I collaborative signals may further concentrate item assignments on a few popular codewords, exacerbating codebook imbalance and collapse. To stabilize the first quantization layer, we introduce Codebook Refresh, a variant of the random-restart strategy [10], as formulated in Eq. 1. We track codeword usage using an exponential moving average (EMA), where $N_{k,t}$ denotes the usage statistic of codeword k , $n_{k,t}$ its assignment count at training step t , and γ the decay factor. When $N_{k,t}$ falls below a threshold τ , the codeword is considered “dead” and reinitialized using the active codebook embedding, with a small Gaussian perturbation. Otherwise, it follows the standard update rule. This process is formulated as:

$$\begin{cases} N_{k,t} = \gamma N_{k,t-1} + (1 - \gamma)n_{k,t}, \\ \mathbf{ce}_{k,t} \leftarrow \mathbb{I}[N_{k,t} \geq \tau] \mathbf{ce}_{k,t}^{\text{update}} + \mathbb{I}[N_{k,t} < \tau] (\mathbf{ce}_{j,t} + \epsilon), \\ \epsilon \sim \mathcal{N}(\mathbf{0}, 0.01^2 \mathbf{I}). \end{cases} \quad (1)$$

where $\mathbf{ce}_{k,t}$ denotes the k -th codeword embedding, $\mathbf{ce}_{k,t}^{\text{update}}$ represents its standard update, and $\mathbf{ce}_{j,t}$ is the j -th codeword embedding which is more active. This mechanism reactivates underutilized codewords and mitigates persistent codebook collapse. We restrict Codebook Refresh to the first quantization layer L_1 , as frequent reinitialization in deeper layers may disrupt residual quantization.

3.1.3 Fidelity Anti-collapse Regularization. Although Codebook Refresh mitigates codebook collapse in the first quantization layer L_1 , codewords in deeper layers may remain underutilized. To address this issue, we introduce Fidelity Anti-collapse Regularization (FAR) from two complementary perspectives. At the sample level, we enable each sample to update multiple semantically relevant codewords instead of only its nearest one, expanding gradient coverage. At the codeword level, we regularize codeword assignments across samples by penalizing excessive collisions, preventing a large number of samples from concentrating on a few codewords.

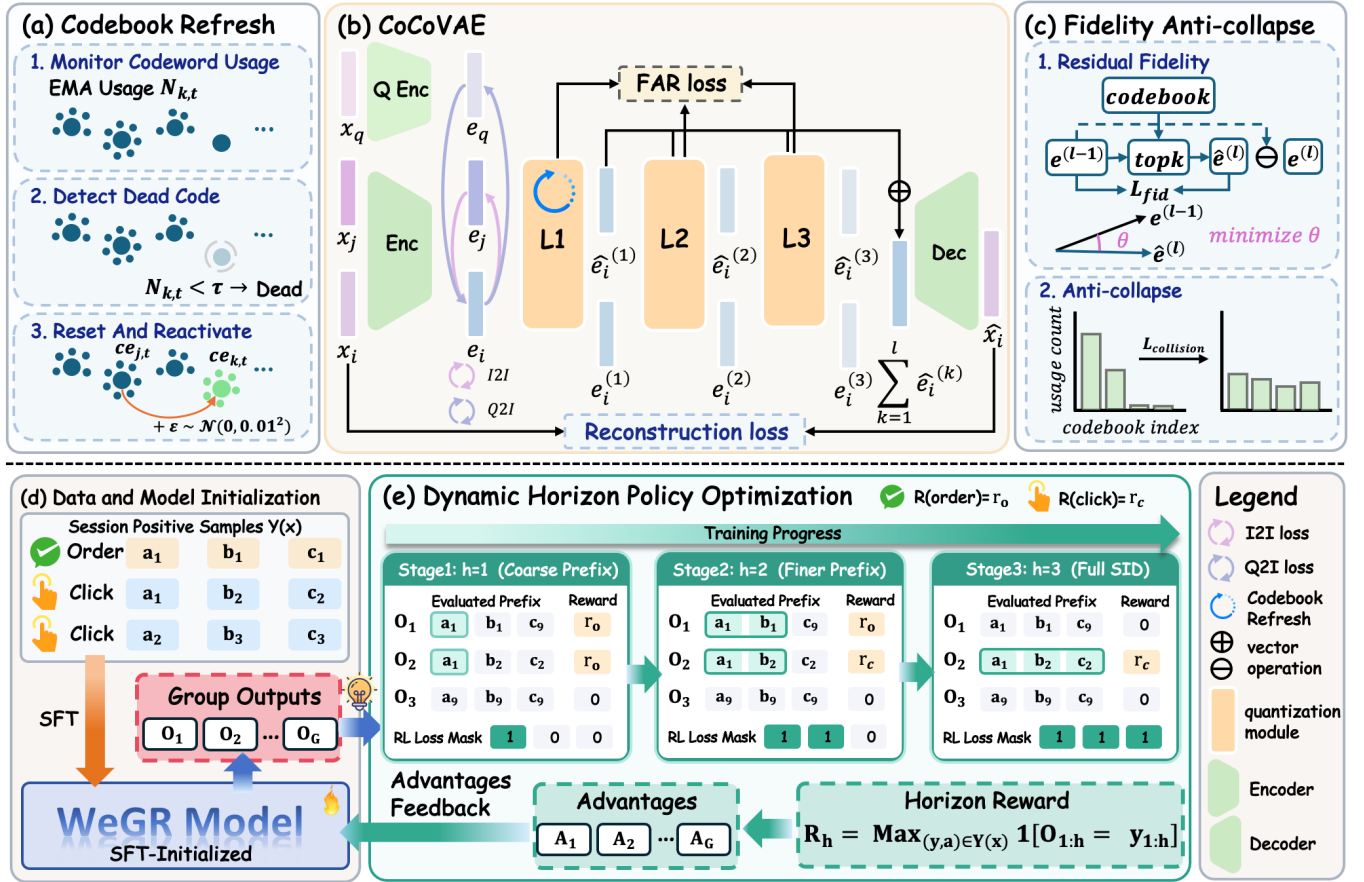


Figure 2: Overview of the proposed WeGR framework. The upper panels (a–c) illustrate CoCo-VAE SID tokenizer construction: (a) codebook refresh reactivates underutilized codewords; (b) Q2I and I2I collaborative signals guide SID learning; and (c) residual fidelity and anti-collapse regularization preserve semantic information and improve codebook utilization. The lower panels (d–e) illustrate preference optimization: (d) training data and model initialization, including session-level positive samples; and (e) DHPO progressively expands reward evaluation and policy optimization horizons, enabling coarse-to-fine preference learning along the SID hierarchy.

Sample-level Residual Fidelity. In conventional RQ-VAE [25], hard top-1 assignment limits codeword updates to the nearest codeword, leaving other semantically relevant codewords insufficiently trained. To address this limitation, we select the most similar codewords for each residual embedding and aggregate them using softmax-normalized similarity weights. Specifically, let $T_i^{(l)}$ denote the set of selected codeword indices for sample i at layer l , where $|T_i^{(l)}| = K_{\text{top}}$, and $w_{ik}^{(l)}$ denote the corresponding aggregation weights. We construct a weighted representation $\hat{e}_i^{(l)}$ and minimize its cosine distance from the original residual embedding $e_i^{(l-1)}$:

$$\begin{cases} \mathcal{L}_{\text{fid}} = \frac{1}{LB} \sum_{l=1}^L \sum_{i=1}^B \left[1 - \cos(e_i^{(l-1)}, \hat{e}_i^{(l)}) \right], \\ \hat{e}_i^{(l)} = \sum_{k \in T_i^{(l)}} w_{ik}^{(l)} \mathbf{ce}_k^{(l)}, \end{cases} \quad (2)$$

where $\mathbf{ce}_k^{(l)}$ denotes the k -th codeword embedding at layer l , while B and L represent the batch size and the number of quantization layers, respectively. This sample-level objective allows multiple relevant

codewords to receive gradient updates simultaneously, improving codebook utilization while preserving residual semantics.

Codeword-level Collision Regularization. While sample-level residual fidelity broadens gradient updates across relevant codewords, it does not explicitly regulate their overall assignment distribution. We therefore introduce a codeword-level Anti-Collapse Loss to prevent different samples from concentrating on the same codewords, promoting more balanced codebook utilization. Specifically, we compute the probability distribution of each sample i over all codewords at quantization layer l , based on the cosine similarities between its residual embedding and the corresponding codewords:

$$p_{ik}^{(l)} = \frac{\exp(s_{ik}^{(l)} / \beta)}{\sum_{h=1}^{M_l} \exp(s_{ih}^{(l)} / \beta)}, \quad (3)$$

where $p_{ik}^{(l)}$ denotes the probability of selecting the k -th codeword for sample i at layer l , $s_{ik}^{(l)}$ represents the cosine similarity between its residual embedding and the corresponding codeword, M_l is the

codebook size at layer l , and β is the softmax temperature. For two distinct samples i and j , we measure their codeword collision probability as the overlap between their assignment distributions:

$$C_{ij}^{(l)} = \langle \mathbf{p}_i^{(l)}, \mathbf{p}_j^{(l)} \rangle = \sum_{k=1}^{M_l} p_{ik}^{(l)} p_{jk}^{(l)}, \quad (4)$$

where $\mathbf{p}_i^{(l)}$ and $\mathbf{p}_j^{(l)}$ represent the probability distributions of samples i and j over all M_l codewords at layer l . A higher collision probability indicates a greater likelihood that both samples select the same codeword.

We then minimize the average collision probability across distinct sample pairs and all quantization layers:

$$\mathcal{L}_{\text{collision}} = \frac{1}{L} \sum_{l=1}^L \frac{1}{B(B-1)} \sum_{i \neq j} C_{ij}^{(l)}, \quad (5)$$

By penalizing codeword collisions across samples, this objective prevents excessive assignments to a few popular codewords, encouraging more balanced codebook utilization.

By combining sample-level residual fidelity with codeword-level collision regularization, we formulate the Fidelity Anti-collapse Regularization (FAR) objective as:

$$\mathcal{L}_{\text{FAR}} = \lambda_{\text{fid}} \mathcal{L}_{\text{fid}} + \lambda_{\text{collision}} \mathcal{L}_{\text{collision}}, \quad (6)$$

Together, these two objectives improve codebook utilization by expanding gradient supervision and reducing codeword concentration across samples.

Overall Objective. The overall training objective combines FAR with the reconstruction loss to preserve item semantic information while mitigating codebook collapse:

$$\mathcal{L}_{\text{all}} = \lambda_{\text{Rec}} \mathcal{L}_{\text{Rec}} + \lambda_{\text{FAR}} \mathcal{L}_{\text{FAR}} + \lambda_{\text{Q2I}} \mathcal{L}_{\text{Q2I}} + \lambda_{\text{I2I}} \mathcal{L}_{\text{I2I}}, \quad (7)$$

where $\mathcal{L}_{\text{Rec}}$ denotes the reconstruction loss, $\mathcal{L}_{\text{Q2I}}$ and $\mathcal{L}_{\text{I2I}}$ denotes the Q2I/I2I InfoNCE loss[22] and the λ coefficients balance the contributions of different objectives.

Together with collaborative regularization and Codebook Refresh, FAR enables CoCo-VAE to preserve behavioral information while improving codebook utilization, yielding more informative and stable SIDs for downstream generative retrieval.

3.2 Dynamic Horizon Policy Optimization

To alleviate reward sparsity in generative retrieval, we propose Dynamic Horizon Policy Optimization (DHPO), a coarse-to-fine reinforcement learning framework exploiting SID hierarchies. Instead of directly optimizing complete SIDs, DHPO progressively expands both reward evaluation and policy optimization horizons, enabling a gradual transition from prefix-level to item-level preference learning without auxiliary reward models.

3.2.1 Dynamic-Horizon Curriculum Learning. For each search session with query q and user context x , we construct a positive set $\mathcal{Y}^+(x) = \{(\mathbf{y}^{(j)}, a^{(j)})\}_{j=1}^{M_x}$ containing the items clicked or ordered by the user, where M_x is the number of positive samples in the session, $\mathbf{y}^{(j)} = (y_1^{(j)}, \dots, y_L^{(j)})$ denotes the L -token SID of a positive item, and $a^{(j)} \in \{\text{click}, \text{order}\}$ denotes the corresponding user action. We define the item-level outcome reward as

$$R_{\text{item}}(\hat{\mathbf{y}}, x) = \max_{1 \leq j \leq M_x} r(a^{(j)}) \cdot \mathbb{I}[\hat{\mathbf{y}} = \mathbf{y}^{(j)}], \quad (8)$$

where $r(\cdot)$ assigns rewards r_o and r_c to orders and clicks, respectively, with $r_o > r_c > 0$.

DHPO divides policy optimization into multiple stages based on the hierarchical structure of SIDs. We introduce a dynamic horizon $h \in \{1, \dots, L\}$, which determines the number of SID tokens used for reward evaluation and policy optimization at each stage. Training starts with $h = 1$ and gradually extends to the complete SID length L , enabling a progressive transition from prefix-level preference learning to item-level ranking optimization. The horizon increases by one after every η optimization steps, where η denotes the number of training steps in each stage. Formally, at optimization step n , the horizon is defined as

$$h(n) = \min \left(1 + \left\lfloor \frac{n}{\eta} \right\rfloor, L \right), \quad (9)$$

At each training stage, we assign behavioral rewards to generated SIDs based on whether their first h tokens match those of clicked or ordered items in the same session:

$$R_h(\hat{\mathbf{y}}, x) = \max_{1 \leq j \leq M_x} r(a^{(j)}) \cdot \mathbb{I}[\hat{\mathbf{y}}_{1:h} = \mathbf{y}_{1:h}^{(j)}]. \quad (10)$$

When multiple positive items share the same prefix, the highest reward is retained. At shorter horizons, generated SIDs can receive rewards by matching only the initial tokens of positive items, even when their complete sequences differ. As h increases, DHPO progressively refines preference learning from shorter SID prefixes to complete item-level matching. Importantly, DHPO preserves the original outcome-based rewards while adjusting the matching granularity, avoiding the need for auxiliary reward models.

Given G sampled rollouts $\{\hat{\mathbf{y}}^{(i)}\}_{i=1}^G$, we compute the group-relative advantage using the rewards at the current horizon:

$$\hat{A}_i = \frac{R_h(\hat{\mathbf{y}}^{(i)}, x) - \text{mean}_{j \in G} (R_h(\hat{\mathbf{y}}^{(j)}, x))}{\text{std}_{j \in G} (R_h(\hat{\mathbf{y}}^{(j)}, x)) + \delta}, \quad (11)$$

where δ is a constant for numerical stability. At each horizon, DHPO compares the rewards of sampled SID prefixes and adjusts their relative generation probabilities to favor those with higher rewards.

Since prefix matching does not guarantee that the complete generated SID corresponds to a positive item, applying the reward to the entire sequence may cause inaccurate supervision for subsequent tokens. DHPO therefore aligns policy optimization with the reward horizon, updating only the first h tokens to ensure accurate credit assignment. The optimization objective is defined as

$$\mathcal{L}_{\text{DHPO}} = -\frac{1}{G} \sum_{i=1}^G \frac{1}{h} \sum_{t=1}^h \min \left(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right), \quad (12)$$

where $\rho_{i,t}(\theta) = \pi_\theta(\hat{y}_t^{(i)} | x, \hat{\mathbf{y}}_{<t}^{(i)}) / \pi_{\theta_{\text{old}}}(\hat{y}_t^{(i)} | x, \hat{\mathbf{y}}_{<t}^{(i)})$ denotes the per-token importance ratio. By jointly expanding the reward and optimization horizons, DHPO gradually extends behavioral supervision to deeper SID positions. When $h = L$, the prefix reward recovers the original item-level outcome reward, i.e., $R_L = R_{\text{item}}$, completing the transition from hierarchical preference learning to exact item-level optimization.

3.2.2 Multi-Positive Supervised Fine-Tuning. Although DHPO alleviates sparse rewards through progressive prefix matching, some

Table 1: Overall performance comparison on the Click and Order datasets. The best results are highlighted in bold.

Method	Click (20k)						Order (20k)					
	HR@6	HR@50	HR@200	MRR@6	MRR@50	MRR@200	HR@6	HR@50	HR@200	MRR@6	MRR@50	MRR@200
<i>Semantic ID Construction (SFT)</i>												
RQ-Kmeans	0.2641	0.4120	0.4869	0.1714	0.1815	0.1823	0.4051	0.5890	0.6613	0.2702	0.2833	0.2841
RQ-Kmeans + L3 Balance	0.2491	0.3746	0.4455	0.1657	0.1745	0.1753	0.3832	0.5374	0.6137	0.2602	0.2711	0.2719
RQ-OPQ	0.2585	0.4000	0.4510	0.1694	0.1790	0.1796	0.3897	0.5661	0.6188	0.2621	0.2744	0.2751
RQ-VAE	0.1865	0.2775	0.3185	0.1216	0.1281	0.1285	0.3470	0.4922	0.5416	0.2313	0.2419	0.2425
R ³ -VAE	0.2149	0.3246	0.3846	0.1432	0.1507	0.1514	0.3505	0.5047	0.5748	0.2343	0.2449	0.2456
CoCo-VAE	0.2759	0.4293	0.5093	0.1821	0.1925	0.1934	0.4103	0.5956	0.6732	0.2776	0.2909	0.2918
<i>Reinforcement Learning (RL)</i>												
DPO	0.2693	0.4107	0.4795	0.1785	0.1882	0.1890	0.4051	0.5741	0.6443	0.2781	0.2899	0.2907
S-DPO (OneSearchV1)	0.2701	0.4200	0.4950	0.1787	0.1891	0.1899	0.4042	0.5845	0.6592	0.2747	0.2874	0.2882
LDPO	0.2725	0.4172	0.4928	0.1792	0.1893	0.1901	0.4121	0.5840	0.6627	0.2792	0.2912	0.2921
Vanilla GRPO	0.2588	0.3934	0.4437	0.1691	0.1786	0.1792	0.3910	0.5576	0.6098	0.2639	0.2758	0.2765
TPMA-GRPO (OneSearchV2)	0.2750	0.4264	0.5067	0.1803	0.1906	0.1915	0.4133	0.5945	0.6722	0.2837	0.2966	0.2974
HCGRec	0.2710	0.4244	0.5045	0.1775	0.1879	0.1888	0.4081	0.5905	0.6664	0.2705	0.2833	0.2842
DHPO (Ours)	0.2810	0.4304	0.5108	0.1851	0.1953	0.1962	0.4217	0.5995	0.6763	0.2878	0.3003	0.3011

rollout groups may still exhibit insufficient reward variation, particularly during early training or for sparse user interactions. Moreover, industrial search naturally involves multiple positive items associated with the same user context, reflecting diverse user interests and behavioral preferences.

To address these challenges, we propose Multi-Positive Supervised Fine-Tuning (MSFT) as a complementary regularization objective. Rather than optimizing a single positive item for each training instance, MSFT jointly supervises all observed positive SIDs:

$$\mathcal{L}_{\text{MSFT}} = -\frac{1}{M_x} \sum_{j=1}^{M_x} \frac{1}{L} \sum_{t=1}^L \log \pi_{\theta} \left(y_t^{(j)} \mid x, \mathbf{y}_{<t}^{(j)} \right). \quad (13)$$

MSFT complements DHPO by jointly supervising multiple relevant items, thereby preserving the model’s ability to capture diverse user interests. Meanwhile, its next-token prediction objective provides stable learning signals across the entire SID sequence, even when sparse rewards lead to degenerate group-relative advantages, helping maintain the retrieval capability established during supervised initialization.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{DHPO}} + \lambda_{\text{MSFT}} \mathcal{L}_{\text{MSFT}}, \quad (14)$$

where λ_{MSFT} controls the contribution of the MSFT objective.

4 Experiment

4.1 Experimental Settings

Datasets. We extract high-quality user interactions from WeChat e-commerce search logs collected between June and July 2026 for supervised fine-tuning (SFT) and reinforcement learning (RL). The dataset spans 31 days, with the first 30 for training and the last for testing. All offline evaluations and ablation studies use this dataset. The Click and Order test sets each contain 20,000 search interactions with observed positive items.

Evaluation Metrics. Following existing generative retrieval studies [2, 3, 9, 31], we evaluate recall and ranking performance using Hit Rate (HR) and Mean Reciprocal Rank (MRR). We report results at $K \in \{6, 50, 200\}$, averaged over all evaluation samples.

Baseline Methods. We compare WeGR with SID construction and preference optimization baselines. SID construction baselines include RQ-Kmeans-based methods, namely RQ-Kmeans [9], RQ-Kmeans with balancing, and RQ-OPQ [2], as well as RQ-VAE-based methods such as RQ-VAE [16, 25, 32] and R³-VAE [31]. For preference optimization, we compare DPO [24], S-DPO [2, 5], and LDPO [12, 28] as off-policy RL baselines, and vanilla GRPO [27], TPMA-GRPO [3], and HCGRec [35] as on-policy RL baselines.

Implementation Details. We adopt GPT-2 Small [23] as the backbone for both offline evaluation and online deployment due to its computational efficiency and low inference latency. For CoCo-VAE, we use three codebook layers with sizes of (16384, 128, 128). The loss weights λ_{fid} , $\lambda_{\text{collision}}$, λ_{FAR} , λ_{Q2I} , λ_{I2I} , and λ_{Rec} are set to 0.9, 1.0, 0.4, 0.2, 0.05 and 1.0, respectively. For RL, we assign rewards of $r_o = 2$ and $r_c = 1$ to orders and clicks, respectively, and set the MSFT weight to $\lambda = 0.5$. We use a GRPO group size of $G = 8$ and $\eta = 300$ optimization steps per DHPO stage, with η adjustable to control stage duration. We update the model daily, performing SFT followed by RL.

4.2 Offline Performance

Table 1 summarizes the performance of SID construction and preference optimization methods on the Click and Order datasets. We analyze the results from two aspects.

Effectiveness of SID Construction. As shown in Table 1, CoCo-VAE consistently outperforms all SID construction baselines across both Click and Order datasets. Compared with RQ-Kmeans, the strongest SFT baseline, CoCo-VAE improves HR@6 by 4.47% and 1.28% on Click and Order, respectively, with further gains of 4.60% and 1.80% at HR@200. CoCo-VAE also substantially outperforms

existing RQ-VAE-based methods, demonstrating its effectiveness across different quantization paradigms. These improvements highlight the importance of jointly preserving collaborative information and maintaining effective codebook utilization. By mitigating codebook collapse while incorporating user interaction signals, CoCo-VAE can better capture behavioral relationships and distinguish different items in the hierarchical SID space, leading to more accurate item representations and improved recall performance.

Effectiveness of Preference Optimization. All RL methods are initialized from CoCo-VAE. As shown in Table 1, existing methods often struggle to improve preference ranking while maintaining retrieval capability. DPO-based methods underperform the SFT baseline, mainly because shared SID prefixes between chosen and rejected items lead to conflicting gradients and weaken learning signals. Moreover, DPO may suffer from reward hacking, increasing preference margins by lowering rejected-item probabilities rather than increasing those of chosen items. For on-policy RL, Vanilla GRPO underperforms SFT across all metrics, as sparse outcome rewards leave many training samples without effective learning signals. TPMA-GRPO improves credit assignment through prefix-matching rewards that do not directly reflect user preferences, while HCGRec uses prefix hints to recover learning signals but may introduce a training-inference mismatch. In contrast, DHPO progressively expands both reward evaluation and policy optimization horizons, reducing learning difficulty while providing reward signals at different levels of the SID hierarchy to guide preference optimization. Compared with TPMA-GRPO, DHPO achieves relative gains of 2.18% and 2.03% in HR@6, and 2.66% and 1.45% in MRR@6 on Click and Order, respectively, demonstrating more effective preference alignment while preserving retrieval capability.

4.3 Ablation Study

In this section, we conduct ablation studies to evaluate the key components of CoCo-VAE and DHPO.

Ablation of CoCo-VAE. To verify the effectiveness of each component in CoCo-VAE, we conduct ablation studies on the Click and Order datasets, as shown in Table 2. We consider four ablation variants: *i*) *w/o q2i* removes query-to-item collaborative supervision; *ii*) *w/o i2i* removes item-to-item collaborative supervision; *iii*) *w/o fid* removes the residual fidelity loss; and *iv*) *w/o coll* removes the anti-collapse regularization. The results demonstrate the complementary roles of these components. Removing *q2i* reduces HR@6 from 0.2759 to 0.2650 on Click and from 0.4103 to 0.3933 on Order, highlighting the importance of query-item co-occurrence in SID construction. Similarly, removing *i2i* reduces HR@6 to 0.2704 and 0.4042 on Click and Order, respectively, indicating the contribution of item-level collaborative relationships. Moreover, the performance degradation caused by removing residual fidelity—with HR@6 decreasing to 0.2437 and 0.3944 on Click and Order, respectively—or anti-collapse regularization—with HR@6 decreasing to 0.2492 and 0.3987—highlights their respective roles in preserving semantic information and promoting effective codebook utilization. Together, these components enable CoCo-VAE to learn informative SID representations for generative retrieval.

Ablation of DHPO. As shown in Table 3, applying vanilla GRPO leads to substantial performance degradation, as sparse outcome

Table 2: Ablation Study of CoCoVAE. The best results are highlighted in bold.

Methods	Click (20k)		Order (20k)	
	HR@6	MRR@6	HR@6	MRR@6
CoCo-VAE	0.2759	0.1821	0.4103	0.2776
<i>w/o q2i</i>	0.2650	0.1732	0.3933	0.2604
<i>w/o i2i</i>	0.2704	0.1748	0.4042	0.2693
<i>w/o fid</i>	0.2437	0.1585	0.3944	0.2602
<i>w/o coll</i>	0.2492	0.1627	0.3987	0.2644

rewards leave many training samples without effective learning signals, undermining the recall capability established during SFT. Introducing MSFT largely mitigates the performance degradation caused by sparse outcome rewards by providing stable supervision over multiple positive items while capturing diverse user interests. Finally, DHPO progressively expands both reward evaluation and policy optimization horizons, enabling coarse-to-fine preference learning from SID prefixes to complete items. Compared with the SFT baseline, DHPO achieves relative improvements of 1.85% and 2.78% in HR@6, and 1.65% and 3.67% in MRR@6 on Click and Order, respectively, demonstrating its effectiveness in preference optimization while preserving retrieval capability.

Table 3: Ablation Study of DHPO. The best results are highlighted in bold.

Methods	Click (20k)		Order (20k)	
	HR@6	MRR@6	HR@6	MRR@6
CoCo-VAE	0.2759	0.1821	0.4103	0.2776
GRPO	0.2588	0.1691	0.3910	0.2639
GRPO + MSFT	0.2764	0.1830	0.4160	0.2835
DHPO	0.2810	0.1851	0.4217	0.2878

4.4 Online A/B Testing

To evaluate the effectiveness and efficiency of WeGR in a production environment, we conducted controlled online A/B tests against the incumbent search system from June 22 to August 4. All results in Table 4 are reported as relative changes over the control group and are statistically significant ($p < 0.05$). We evaluate three deployment configurations that differ in the extent to which WeGR replaces candidates from conventional retrieval channels. **V1.0 (Additive Retrieval)** introduces WeGR as an additional retrieval channel while retaining all candidates from the other retrievers. **V1.1 (60% Candidate Removal)** and **V1.2 (80% Candidate Removal)** remove 60% and 80% of the candidates contributed by the other retrieval channels, respectively. These configurations allow us to examine whether the business gains persist or improve as WeGR progressively replaces conventional retrieval candidates.

Table 4: Online A/B testing results from June 22 to August 4. All numbers denote relative improvements over the production baseline, with bold values indicating statistical significance ($p < 0.05$).

Deployment Configuration	Order	CTR	CVR
V1.0 (Additive Retrieval)	+1.579%	+1.108%	+0.520%
V1.1 (60% Candidate Removal)	+1.962%	+0.884%	+0.950%
V1.2 (80% Candidate Removal)	+2.539%	+1.508%	+1.381%

As shown in Table 4, all three deployment configurations yield statistically significant improvements in Order, CTR, and CVR ($p < 0.05$). Under the additive configuration, V1.0 increases Order, CTR, and CVR by 1.579%, 1.108%, and 0.520%, respectively. After removing 60% of the candidates contributed by the other retrievers, V1.1 further increases the Order and CVR gains to 1.962% and 0.950%, while maintaining a significant CTR improvement of 0.884%. When the removal ratio reaches 80%, V1.2 yields the largest gains across all three metrics, improving Order by 2.539%, CTR by 1.508%, and CVR by 1.381%. These results show that the business gains do not depend on enlarging the candidate set; instead, the overall performance further improves as WeGR replaces a larger proportion of candidates from conventional retrieval channels.

Online resource savings. Beyond the improvements in business metrics, WeGR substantially reduces online serving costs. The key reason is that WeGR replaces, rather than simply supplements, a large fraction of candidates produced by conventional retrieval channels. In V1.2, cutting off 80% of the candidates from the other retrievers reduces the number of items entering pre-ranking to 61.90% of the baseline, i.e., 38.10% fewer items need to be scored. Since pre-ranking performs item-wise feature processing and model inference over the merged candidate set, this reduction directly decreases its CPU and GPU workloads. As shown in Figure 3, the normalized CPU and GPU usages decrease from 100% to 74.5% and 73.9%, corresponding to resource savings of 25.5% and 26.1%, respectively. The resource reduction is smaller than the reduction in scored items because the online pipeline still incurs fixed overhead from retrieval, request processing, and other serving components. These results demonstrate that WeGR improves online effectiveness while alleviating computation cost of the downstream cascade.

Online latency reduction. WeGR also improves serving efficiency across two production search scenarios. As reported in Table 5, it reduces the average online latency by 21.7 ms in WeChat Channels Search and by 15.4 ms in WeChat Search. These consistent latency reductions are primarily attributable to the reduced number of candidates passed to the pre-ranking stage, thereby lowering the computational overhead of per-item feature processing and model inference. Together with the online A/B testing gains, the results demonstrate that WeGR improves business performance while reducing serving latency in industrial search systems.

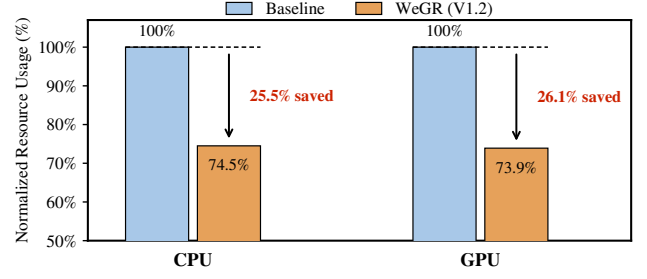


Figure 3: The comparisons of online resource savings for baseline and WeGR.

Table 5: Average online latency reductions achieved by WeGR.

Production Scenario	Latency Reduction
WeChat Channels Search	21.7 ms
WeChat Search	15.4 ms

4.5 Further Analysis

In this section, we investigate some questions regarding long-term behavior and user-level benefits of WeGR in industrial deployment.

1) **How does the performance of WeGR evolve over time?** Industrial search systems operate over a continuously changing item corpus and user interests. It is therefore important to examine whether the retrieval quality of WeGR remains stable as the system evolves. Figure 4 reports the daily Click and Order HR@200 from June 19 to August 19, 2026. Despite short-term fluctuations, both metrics exhibit a clear upward trend. Specifically, the average Click HR@200 increases from 69.98% in the first seven days to 76.99% in the last seven days, while the corresponding Order HR@200 rises from 80.24% to 87.29%. The Order HR@200 remains consistently higher than the Click HR@200 throughout the observation period, indicating that WeGR effectively retrieves high-value items associated with downstream conversions. These results demonstrate that WeGR maintains robust retrieval quality over time and benefits from continuously accumulated interaction signals rather than degrading as the item corpus and user interests evolve.

2) **Which user groups contribute most to the online gains of WeGR?** We divide users into six groups according to their lifecycle status: active existing users (UG1), reactivated existing users (UG2), lapsed existing users (UG3), active new users (UG4), reactivated new users (UG5), and lapsed new users (UG6). As shown in Table 6, WeGR yields positive Order, CTR, and CVR gains across all six groups, demonstrating that its online improvements are not driven by a single user cohort. The gains are particularly pronounced among reactivated and lapsed new users. UG5 achieves the largest improvements in Order and CTR, reaching 9.855% and 3.448%, respectively, while UG6 obtains the highest CVR gain of 7.382%. UG4 also exhibits substantial increases of 4.823% in Order and 3.063% in CVR. In contrast, the gains for active existing users are relatively

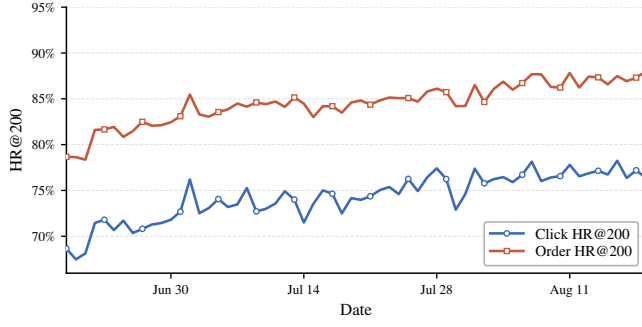


Figure 4: Daily Click and Order HR@200 of WeGR over time.

moderate. This pattern suggests that WeGR is especially effective for users with limited or rapidly changing interaction histories, for whom conventional retrieval methods may suffer from sparse or stale behavioral signals. By jointly exploiting semantic representations and user interactions within a unified generative framework, WeGR can better recover their current search intent and retrieve items with stronger conversion potential.

Table 6: Online relative gains across different user groups. All numbers denote relative improvements over the baseline, and bold values indicate statistical significance ($p < 0.05$).

Metric	UG1	UG2	UG3	UG4	UG5	UG6
Order	+1.611%	+5.154%	+1.556%	+4.823%	+9.855%	+6.143%
CTR	+1.125%	+2.323%	+2.112%	+1.638%	+3.448%	+0.524%
CVR	+0.399%	+2.481%	+0.247%	+3.063%	+6.585%	+7.382%

3) **On which query-frequency regimes does WeGR work best?** We stratify queries by historical frequency into low (0–3), medium-low (4–99), medium (100–999), and high-frequency (≥ 1000) buckets. As shown in Table 7, the strongest business-value gains arise at the long-tail and head regions, while the engagement benefit extends to medium-frequency traffic. On long-tail traffic, WeGR significantly improves Order by 3.24% for low-frequency queries and improves both Order and CVR by 3.86% and 3.58%, respectively, for medium-low-frequency queries. These conversion-oriented gains suggest that the learned SID space can transfer collaborative and semantic structure to queries for which direct behavioral evidence is limited. For high-frequency queries, WeGR yields significant gains of 3.51% in Order and 2.73% in CTR, indicating that the model also effectively exploits abundant interaction signals for head traffic. Medium-frequency queries exhibit a significant CTR improvement of 0.99%, despite statistically insignificant changes in Order and CVR, suggesting that increased engagement does not yet translate into downstream conversion within this bucket. Overall, WeGR exhibits complementary effects across the traffic distribution: it primarily strengthens conversion outcomes in the long tail, improves both engagement and orders at the head, and provides a more moderate engagement benefit in the middle. We emphasize that this analysis identifies where the online gains arise; establishing the precise mechanism behind these heterogeneous treatment effects requires further controlled investigation.

Table 7: Online relative gains across query-frequency buckets. All numbers denote relative improvements over the baseline, and bold values indicate statistical significance ($p < 0.05$).

Query bucket	Frequency	Order	CVR	CTR
Low	0–3	+3.24%	+2.40%	+0.95%
Medium-low	4–99	+3.86%	+3.58%	+0.91%
Medium	100–999	−0.05%	−0.84%	+0.99%
High	≥ 1000	+3.51%	+1.12%	+2.73%

4) **Does DHPO generalize across different industrial search scenarios?** We assess the cross-scenario transferability of DHPO through an independent evaluation in WeChat e-commerce live-stream search, with its SFT-initialized counterpart serving as the reference. As reported in Table 8, DHPO yields consistent improvements across all twelve metrics on the 18K order- and click-supervised evaluation sets. The gains are particularly pronounced at shallow retrieval depths: HR@6 increases by 7.07% and 5.95%, while MRR@6 increases by 7.82% and 6.12%, under order and click supervision, respectively. Moreover, the persistent MRR improvements across cutoffs indicate that DHPO refines the relative ordering of behaviorally preferred items throughout the retrieved list, whereas the larger gains at $K = 6$ suggest a stronger concentration of such items near the top ranks. The consistency of these effects in a distinct production domain supports the central premise of DHPO: preference alignment can be transferred via behavior-grounded, dynamically expanded optimization horizons without a scenario-specific reward model. As this study isolates the policy-optimization stage, the evidence pertains specifically to the transferability of DHPO rather than that of the complete WeGR framework.

Table 8: DHPO performance on WeChat live-stream search relative to SFT.

Metric	Order (18K)			Click (18K)		
	SFT	DHPO	Improve	SFT	DHPO	Improve
HR@6	0.2827	0.3027	+7.07%	0.1982	0.2100	+5.95%
HR@100	0.5237	0.5362	+2.39%	0.4804	0.4935	+2.73%
HR@200	0.5648	0.5762	+2.02%	0.5430	0.5545	+2.12%
MRR@6	0.2045	0.2205	+7.82%	0.1340	0.1422	+6.12%
MRR@100	0.2176	0.2333	+7.22%	0.1475	0.1561	+5.83%
MRR@200	0.2179	0.2336	+7.21%	0.1480	0.1565	+5.74%

5 Conclusion

In this work, we presented WeGR, a unified generative retrieval framework for industrial search systems in WeChat. By combining collaborative-aware SID construction with progressive preference optimization, WeGR effectively improves retrieval and ranking performance while addressing codebook collapse and reward sparsity. Extensive offline experiments and large-scale online A/B tests demonstrate its effectiveness in improving business outcomes and reducing serving costs, highlighting the potential of generative retrieval to simplify industrial search systems. Furthermore, evaluations in an independent live-stream search scenario demonstrate the applicability of DHPO across different industrial search settings.

References

- [1] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. In *Advances in Neural Information Processing Systems*, Vol. 35. Neural Information Processing Systems Foundation, Inc., 31668–31683. doi:10.52202/068431-2296
- [2] Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, Huangyu Dai, Xing Xu, Tong Zhao, Mingcan Peng, Xiaoyang Zheng, Chao Wang, Qihang Zhao, Zhixin Zhai, Yang Zhao, Bocho Liu, Jingshan Lv, Xiao Liang, Yuqing Ding, Jing Chen, Chenyi Lei, Wenwu Ou, Han Li, and Kun Gai. 2025. OneSearch: A Preliminary Exploration of the Unified End-to-End Generative Framework for E-commerce Search. doi:10.48550/arXiv.2509.03236 arXiv:2509.03236
- [3] Ben Chen, Siyuan Wang, Yufei Ma, Zihan Liang, Xuxin Zhang, Yue Lv, Ying Yang, Huangyu Dai, Lingtao Mao, Tong Zhao, Zhipeng Qian, Xinyu Sun, Zhixin Zhai, Yang Zhao, Bocho Liu, Jingshan Lv, Xiao Liang, Hui Kong, Jing Chen, Han Li, Chenyi Lei, Wenwu Ou, and Kun Gai. 2026. OneSearch-V2: The Latent Reasoning Enhanced Self-distillation Generative Search Framework. doi:10.48550/arXiv.2603.24422 arXiv:2603.24422
- [4] Jinwen Chen, Shiwen Zhang, Shuai Gong, Zheng Zhang, Yachao Zhao, Lingxiang Wang, Haibo Zhou, Wei Lin, and Hainan Zhang. 2026. LocalSUG: City-Preference-Enhanced LLM for Query Suggestion in Local-Life Services. doi:10.48550/arXiv.2603.04946 arXiv:2603.04946
- [5] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. 2024. On softmax direct preference optimization for recommendation. *Advances in Neural Information Processing Systems* 37 (2024), 27463–27489.
- [6] Zhiguo Chen, Guohao Sun, Yiming Qiu, Xingzhi Yao, Mingming Li, Huimu Wang, Yangqi Zhang, Songlin Wang, and Sulong Xu. 2026. RAD-DPO: Robust Adaptive Denoising Direct Preference Optimization for Generative Retrieval in E-commerce. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 4507–4512. doi:10.1145/3805712.3808512
- [7] Keertana Chidambaram, Sanath Kumar Krishnamurthy, Qiuling Xu, Ko-Jen Hsiao, and Moumita Bhattacharya. 2026. Robust Post-Training for Generative Recommenders: Why Exponential Reward-Weighted SFT Outperforms RLHF. doi:10.48550/arXiv.2603.10279 arXiv:2603.10279 [cs.LG]
- [8] Xiaopeng Chu, Jianbo Zhu, Mingmin Jin, Jing Wang, Xing Fang, and Wenyi Zhang. 2026. VARG: Value-Aware and Ranking-Aligned Generative Retrieval for Dynamic E-commerce Search. arXiv:2609.14493 <https://arxiv.org/abs/2609.14493>
- [9] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. doi:10.48550/arXiv.2502.18965 arXiv:2502.18965
- [10] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A Generative Model for Music. doi:10.48550/arXiv.2005.00341 arXiv:2005.00341
- [11] Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. 2026. Differentiable Semantic ID for Generative Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 369–379. doi:10.1145/3805712.3809641
- [12] Kairui Fu, Changfa Wu, Kun Yuan, Binbin Cao, Dunxian Huang, Yuliang Yan, Junjun Zheng, Jianning Zhang, Silu Zhou, Jian Wu, and Kun Kuang. 2026. RankGR: Rank-Enhanced Generative Retrieval with Listwise Direct Preference Optimization in Recommendation. doi:10.48550/arXiv.2602.08575 arXiv:2602.08575
- [13] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized Product Quantization. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 4 (2014), 744–755. doi:10.1109/TPAMI.2013.240
- [14] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. ACM, 299–315. doi:10.1145/3523227.3546767
- [15] Kaifeng Guo, Yiming Yang, Jingtong Gao, Guolei Zeng, Fukang Yang, Yukang Liang, Peng Jiang, Qingpeng Cai, and Xiangyu Zhao. 2026. Hierarchical Residual Policy Optimization for Generative Recommendations. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. ACM, 1427–1438. doi:10.1145/3770855.3818206
- [16] Xian Guo, Ben Chen, Siyuan Wang, Ying Yang, Chenyi Lei, and et al. 2025. OneSug: The Unified End-to-End Generative Framework for E-commerce Query Suggestion. (2025). doi:10.48550/arXiv.2506.06913 arXiv:2506.06913
- [17] Doyup Lee, Chihyeon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive Image Generation using Residual Quantization. arXiv:2203.01941 [cs.CV] <https://arxiv.org/abs/2203.01941>
- [18] Mingming Li, Huimu Wang, Zuxu Chen, Guangtao Nie, Yiming Qiu, and et al. 2024. Generative Retrieval with Preference Optimization for E-commerce Search. arXiv:2407.19829 [cs.IR] <https://arxiv.org/abs/2407.19829>
- [19] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2025. Generative Recommender with End-to-End Learnable Item Tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 729–739. doi:10.1145/3726302.3729989
- [20] Xinchun Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2024. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. *arXiv preprint arXiv:2411.11739* (2024).
- [21] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite Scalar Quantization: VQ-VAE Made Simple. arXiv:2309.15505 [cs.CV] <https://arxiv.org/abs/2309.15505>
- [22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
- [23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
- [24] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems* 36 (2023), 53728–53741.
- [25] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., 10299–10315. doi:10.52202/075280-0452
- [26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. doi:10.48550/arXiv.1707.06347 arXiv:1707.06347
- [27] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. doi:10.48550/arXiv.2402.03300 arXiv:2402.03300
- [28] Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 18990–18998.
- [29] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer Memory as a Differentiable Search Index. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, 21831–21843. doi:10.52202/068431-1587
- [30] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. Neural Discrete Representation Learning. arXiv:1711.00937 [cs.LG] <https://arxiv.org/abs/1711.00937>
- [31] Qiang Wan, Ze Yang, Dawei Yang, Ying Fan, Xin Yan, Siyang Liu, Yicong Liu, Chenwei Zhang, Wei Xu, Jiahao Qin, and Ke Wang. 2026. R3-VAE: Reference Vector-Guided Rating Residual Quantization VAE for Generative Recommendation. doi:10.48550/arXiv.2604.11440 arXiv:2604.11440
- [32] 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*. ACM, 2400–2409. doi:10.1145/3627673.3679569
- [33] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A Neural Corpus Indexer for Document Retrieval. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 25600–25614. doi:10.52202/068431-1856
- [34] Jun Zhang, Yi Li, Yue Liu, Changping Wang, Yuan Wang, Yuling Xiong, Xun Liu, Haiyang Wu, Qian Li, Enming Zhang, Jiawei Sun, Xin Xu, Zishuai Zhang, Ruoran Liu, Suyuan Huang, Zhaoxin Zhang, Zhengkai Guo, Shuojin Yang, Meng-Hao Guo, Huan Yu, Jie Jiang, and Shi-Min Hu. 2026. GPR: Towards a Generative Pre-trained One-Model Paradigm for Large-Scale Advertising Recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. doi:10.1145/3770855.3818425
- [35] Kangning Zhang, Haotian Fang, Xukun Luo, Hao Yin, Yang Gao, Peng Yan, Weiwen Liu, Weinan Zhang, and Yong Yu. 2026. Learning from Unreachable Rewards: Hint-Conditioned Reinforcement Learning for Generative Recommendation. doi:10.1145/3799682.3840842 arXiv:2608.11980
- [36] Kun Zhang, Jingming Zhang, Wei Cheng, Yansong Cheng, Jiaqi Zhang, Hao Lu, Xu Zhang, Haixiang Gan, Jiangxia Cao, Tenglong Wang, Ximing Zhang, Boyang Xia, Kuo Cai, Shiyao Wang, Hongjian Dou, Jinkai Yu, Mingxing Wen, Qiang Luo, Dongxu Liang, Chenyi Lei, Jun Wang, Runan Liu, Zhaojie Liu, Ruiming Tang, Tingting Gao, Shaoqun Liu, Yuqing Ding, Hui Kong, Han Li, Guorui Zhou, Wenwu Ou, and Kun Gai. 2026. OneMall: One Architecture, More Scenarios — End-to-End Generative Recommender Family at Kuaishou E-Commerce. doi:10.48550/arXiv.2601.21770 arXiv:2601.21770
- [37] Xiao Zhang, Guanyu Chen, Boyang Zuo, Feng Li, Pengjie Wang, Jian Xu, and Bo Zheng. 2026. Value: Value-aware large language model for query rewriting

- via weighted trie in sponsored search. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 2584–2593.
- [38] 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. doi:10.1109/ICDE60146.2024.00118
- [39] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchun Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfeng Zhao, Zhixin Ling, and Ziming Li. 2025. OneRec-V2 Technical Report. arXiv:2508.20900 [cs.LG] <https://arxiv.org/abs/2508.20900>
- [40] Junjie Zhou, Ke Mei, Lei Li, Tianyi Wang, Fengyun Rao, and Jing Lyu. 2026. WeMM-Embedding: WeChat Multi-Modal Embedding Technical Report. doi:10.48550/arXiv.2608.24053 arXiv:2608.24053
- [41] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing Generative Retrieval with Reinforcement Learning from Relevance Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 12481–12490. doi:10.18653/v1/2023.emnlp-main.768