

Unifying Generative Recall and Multi-Objective Ranking in a Single Decoder-Only Sequence

Ruochen Yang
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, University
of Chinese Academy of Sciences
Beijing, China
yangruochen@iie.ac.cn

Shuang Wen
Kuaishou Technology
Beijing, China
wenshuang@kuaishou.com

Pengbo Xu
Kuaishou Technology
Beijing, China
xupengbo@kuaishou.com

Yusheng Huang
Kuaishou Technology
Beijing, China
huangyusheng@kuaishou.com

Jiangxia Cao*
Kuaishou Technology
Beijing, China
caojiangxia@kuaishou.com

Shuang Yang
Kuaishou Technology
Beijing, China
yangshuang08@kuaishou.com

Zhaojie Liu
Kuaishou Technology
Beijing, China
zhaotianxing@kuaishou.com

Jiawei Sheng
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, University
of Chinese Academy of Sciences
Beijing, China
shengjiawei@iie.ac.cn

Tingwen Liu*
Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, University
of Chinese Academy of Sciences
Beijing, China
liutingwen@iie.ac.cn

Abstract

Modern industrial recommendation systems typically separate recall and ranking into two independent stages. Although this cascade supports corpus-level retrieval and fine-grained multi-objective scoring, it causes objective inconsistency, information loss at the candidate hand-off, and redundant user-side context computation. Meanwhile, the generative recall and ranking scaling share a common Transformer-based modeling philosophy, where architectural consistency creates a natural opportunity for unified integration. However, direct sharing remains challenging since the two tasks require different information visibility and optimization methods. Therefore, we propose **UniR²**, a **Unified** decoder-only Transformer that unifies Generative Recall and Multi-Objective Ranking within a single heterogeneous sequence comprising user context, SID trajectory, and item features. Within this sequence, the generated trajectory serves as a representation bridge between recall and ranking, where Dual-Query Prefix-Causal Attention provides task-specific visibility. The two tasks share the base attention weights but retain separate optimization boundaries, with ranking-side LoRA preserving ranking adaptability without disrupting the generative backbone. Extensive offline experiments on large-scale industrial data demonstrate the effectiveness and efficiency of UniR² for both recall and ranking. Long-term online A/B tests on Kuaishou platform further show consistent positive gains, validating the practicality of unified model in large-scale recommendation systems.

CCS Concepts

• Information systems → Recommender systems.

*Corresponding author.

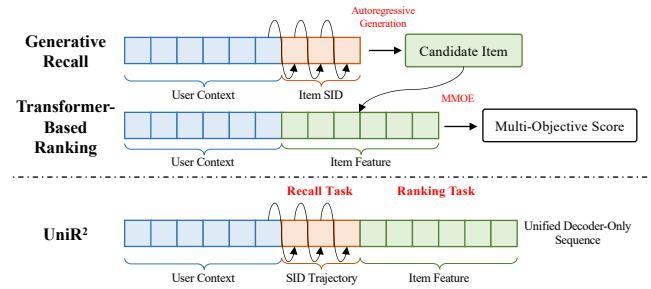


Figure 1: Architecture comparison between cascaded architecture and UniR². Serial generation and scoring with separate models lead to misaligned objectives and redundant user context computation. UniR² unifies both tasks within a single heterogeneous decoder-only sequence, enabling representation reuse and bridging the cross-stage gap.

Keywords

Generative Recommendation, Multi-Objective Ranking

1 Introduction

Modern industrial recommender systems are commonly organized as a multi-stage cascade. Given a user request, the recall stage [2, 31] first retrieves thousands of candidates from a corpus containing billions of items, after which a ranking model [16, 30] performs fine-grained user-item matching under multiple business objectives. This division of labor has supported large-scale recommendation for years, because recall emphasizes efficient search and coverage

whereas ranking emphasizes accurate per-candidate discrimination. However, the separation also introduces a fundamental gap between the two stages. Recall and ranking are usually optimized with different objectives, consume differently organized features, and produce representations in independent parameter spaces. As a result, information captured during candidate generation is compressed into a discrete candidate list at the stage boundary, while user context are repeatedly encoded by different models. These issues lead to objective inconsistency, representation loss, and redundant computation in both training and serving.

Meanwhile, both stages of this cascade are undergoing a similar architectural update toward Transformers [23]. On the recall side, generative recommendation represents each item with a hierarchical semantic ID (SID) and casts large-scale retrieval as autoregressive next-token prediction [4, 18]. The Transformer learns a distribution over the item corpus by generating an SID trajectory conditioned on the user’s behavioral context, thereby avoiding exhaustive item-by-item scoring. On the ranking side, recent models increasingly tokenize heterogeneous fields, behavioral sequences, and contextual features, and use multi-layer attention as a scalable feature interaction backbone [34, 37]. Although the two tasks retain different outputs, they now share several important computational primitives: tokenized inputs, user-context modeling, attention-based feature fusion, and deep Transformer stacks. This convergence naturally raises a question: *Can generative recall and multi-objective ranking be formulated into one Transformer sequence rather than two isolated models?*

Such a unified Transformer offers several potential benefits. First, a common sequence space can preserve the information produced during recall and directly expose it to ranking, reducing the semantic gap at the cascade boundary. Besides, the two tasks can share user-side context computation, since reusing intermediate key/value caches across candidate generation and scoring eliminates the duplicated encoding of long histories and personalized profiles. Furthermore, scaling such a single Transformer backbone allows recall and ranking to benefit jointly, instead of independently maintaining two increasingly large models.

Nevertheless, directly combining the two tasks is non-trivial. Generative recall is a structured distribution fitting problem, where each SID token must obey autoregressive causality and preserve the hierarchical semantics. Multi-objective ranking is instead a discriminative feature interaction problem, where all item-side features are available simultaneously, and each candidate must be evaluated under dense and sparse objectives such as click, long-view, and gift. Therefore, the two tasks require different information visibility and optimization methods. Simply sharing all parameters in a single sequence will lead to a severe seesaw effect between retrieval quality and ranking accuracy.

Existing attempts at recall and ranking collaboration do not fully resolve these issues. Some methods append a lightweight scorer after generation or transfer representations between otherwise separate modules [6, 19]; others share a Transformer encoder or unify the input format while retaining task-specific computation paths [3, 12, 14]. These approaches improve cross-stage collaboration, but the generated SID trajectory and fine-grained ranking

features are still not organized within a single sequence for layer-wise interaction. Consequently, ranking either consumes a compressed post-generation representation or requires an additional feature-fusion network, while the user context may still be redundantly processed. The key challenge is therefore not simply to share an encoder or a loss function, but to construct a single sequence that provides representation coupling in the forward pass while maintaining optimization isolation between the two objectives.

To address this challenge, we propose **UniR²**, a **Unified** decoder-only Transformer that performs **Generative Recall** and **Multi-Objective Ranking** within a single heterogeneous sequence. UniR² organizes user context, SID trajectory, and item features in one Transformer trunk, allowing ranking to consume the recall decision process rather than only its discrete candidate output. A Dual-Query Prefix-Causal Attention mechanism provides task-specific visibility for causal generation and ranking feature fusion, while the two-stage different optimization objectives preserves ranking adaptability without destabilizing the generative backbone. UniR² therefore achieves representation coupling and optimization isolation within one training forward pass, as well as one single service for supporting recall and ranking during online inference.

Our main contributions are summarized as follows:

- We introduce a single-sequence unification perspective for generative recall and multi-objective ranking. By organizing user context, SID trajectory, and item features within one decoder-only sequence, the framework turns the recall trajectory into a representation bridge for layer-wise ranking feature interaction.
- We design UniR², a unified architecture equipped with DQ-PCA and LoRA-balanced optimization isolation. UniR² couples the two tasks in forward representation learning while preventing ranking gradients from corrupting the autoregressive recall backbone, and supports efficient single-service inference deployment through shared context computation.
- We conduct extensive experiments on large-scale industrial data. UniR² consistently improves both generative recall and multi-objective ranking over deployed baselines, and achieves significant gains in online A/B test on real-world Kuaishou services, demonstrating its effectiveness and practicality for industry.

2 Related Work

2.1 Generative Recommendation

Generative recommendation reformulates sequential recommendation as a subsequent SID prediction and generation task conditioned on users’ historical interaction sequences. It enables direct hierarchical search from user context to massive item candidates without scoring the entire item corpus one by one, making it naturally suitable for large-scale recall in industrial recommender systems. This paradigm was first established by TIGER [18] with an item tokenizer and an encoder-decoder architecture, and was later extended by the OneRec series [4, 26, 29, 33], which introduced reinforcement learning for personalized preference alignment. However, the simple autoregressive generation and beam search does not fully address the need to directly compare candidates and explore higher value ones. PROMISE [7] introduces a process reward model to perform fine-grained evaluation of intermediate generation paths. RankGR [6] refines candidates with a lightweight scoring model.

V-Star [11] focuses the search on critical SID branches through a structured sampling strategy. Nevertheless, these methods essentially rely on a single preference signal and do not explicitly model candidates' multi-objective scores required in the ranking stage.

2.2 Transformer-Based Ranking

Industrial ranking models have long relied on multi-task tower architectures [16, 21, 27], achieving strong performance through manually designed feature-interaction mechanisms [25, 36]. However, modeling user interests from different perspectives with independent networks often leads to shallow interactions among heterogeneous information, fragmented computation graphs, and limited scalability of model capacity. In recent years, inspired by Transformer architectures [23], industrial ranking models have started to represent discrete fields, contextual features, and user behavior sequences uniformly as tokens. Multi-layer attention is then used to simultaneously perform cross-field feature interaction and interest modeling, gradually endowing ranking models with an LLM-like unified backbone and scalable model capacity [1, 32]. OneTrans [34] jointly models sequential and non-sequential features through a Transformer backbone. MixFormer [10] co-scales dense feature interaction and sequence modeling within a single parameter space. SORT [24] significantly improves the computational efficiency of ranking Transformer models through system-level optimization. TokenFormer [37] further addresses the issue of sequence representation collapse, enhancing the discriminability of multi-field feature representations while preserving sequence modeling capability. Nevertheless, these methods still operate only at the ranking stage. Since they share the Transformer-based modeling philosophy with generative recall, there remains substantial potential to bridge recall and ranking within a unified model.

2.3 Unified Recall and Ranking

Collaborative modeling of recall and ranking has recently emerged as an industry-relevant research direction, evolving from objective alignment and parameter sharing toward the integration of models and computation pipelines. Early work such as RankGR [6] introduces a candidate scoring module into generative recall and jointly optimizes sparse parameters, yet the two stages remain separated at the model level, which in essence is mapping recall and ranking into the same representation space. Subsequent studies have increasingly explored unified architectures. OnePiece [3] applies unified tokenized inputs, Transformer architectures, and contextual reasoning to both recall and ranking. OneRanker [19] achieves coarse-to-fine collaborative optimization through key/value transfer between generation and ranking, together with distribution-consistency constraints. UniPinRec [12] unifies input formats and model training by sharing a Transformer. However, existing methods typically rely on post-generation scoring or cross-module information transfer, and have not yet organized recall trajectories and fine-grained ranking features within the same sequence for layer-wise interaction. In contrast, while sharing the backbone network between recall and ranking objectives, UniR² integrates user context encoding, SID generation trajectories, and feature interactions into a single decoder sequence, achieving unification at both the token-sequence level and the layer-wise computation level.

3 Preliminary

Let $\mathcal{U}$ and $\mathcal{V}$ denote the user and item sets. Each user $u \in \mathcal{U}$ is associated with personalized profile and multiple history sequences of diverse behaviors. Besides, each item $v \in \mathcal{V}$ is described by a set of raw features, containing semantic ID (SID) $s_v = \{q_1, q_2, \dots, q_L\}$ obtained through discrete quantization [4, 18, 26], where q_l is the code of the l -th layer. We employ Res-Kmeans [15] with $L = 3$ layers and codebook size 8129 in our scenario. For every observed interaction pair (u, v) , we collect a multi-objective label vector $\mathbf{y} = (y_1, \dots, y_T) \in \{0, 1\}^T$ (e.g., click, long-view, gift) based on whether the corresponding behavior occurred.

Therefore, our goal is to accomplish two tasks on the same model foundation. **Generative Recall** task learns an autoregressive distribution over SIDs conditioned on the user-side context:

$$p_\theta(s_v | u) = \prod_{i=1}^L p_\theta(q_i | u, q_{<i}). \quad (1)$$

NTP loss is adopted during training, while beam search is performed at serving time. **Multi-Objective Ranking** task then learns per-objective probabilities for a given (u, v) pair. Formally, we seek a score function:

$$\hat{y} = f_\phi(u, v) \in [0, 1]^T, \quad (2)$$

whose t -th coordinate $\hat{y}_t$ estimates $P(y_t = 1 | u, v)$. The training of ranker is based on multi-objective BCE loss.

Traditional cascade methods instantiate the two tasks with independent parameter sets θ and ϕ , and pass the candidate list produced by the generation task to the ranking task for multi-objective estimation. In this paper, we study an alternative formulation that learns a single model parameterized by Θ that can produce both the generative distribution and the ranking scores within a single stage of the pipeline. In this way, the candidates and their corresponding scores can be obtained in one service pass.

4 Methodology

UniR² is a single decoder-only transformer that unifies generative recall and multi-objective ranking in one sequence, thereby maximizing model reuse and reducing computational redundancy. As illustrated in Figure 2, the input of UniR² is a sequentially arranged heterogeneous token sequence encompassing both user and item side information. Through a carefully designed strategy, the outputs of different segments are respectively dedicated to the generation and ranking tasks.

4.1 Unified Sequence Composition

For each observable (u, v) interaction pair, all relevant features are available during training. We therefore assemble these features into a single unified sequence $\mathcal{S} = [\mathcal{P} \parallel \mathcal{T}_{\text{gen}} \parallel \mathcal{T}_{\text{rank}}]$ that concatenates three groups of tokens as the model input.

User segment $\mathcal{P}$. We view u 's side information as a set of heterogeneous behavioral and profile blocks $\{\mathbf{x}_u^{(0)}, \mathbf{x}_u^{(1)}, \dots, \mathbf{x}_u^{(K-1)}, \mathbf{x}_u^{(K)}\}$, where $\mathbf{x}_u^{(K)}$ encodes u 's static profile and $\mathbf{x}_u^{(b)}$ ($0 \leq b \leq K-1$) is a variable-length interaction history sequence under behavior $b \in \{\text{Click, Long-View, Gift, } \dots\}$. Each block is projected to d and shifted by its own type embedding drawn from a table $E_{\mathcal{P}} \in \mathbb{R}^{(K+1) \times d}$:

$$\mathcal{P}_k = \text{Proj}_k(\mathbf{x}_u^{(k)}) + E_{\mathcal{P}}[k], \quad \mathcal{P} = [\mathcal{P}_0; \mathcal{P}_1; \dots; \mathcal{P}_{K-1}; \mathcal{P}_K]. \quad (3)$$

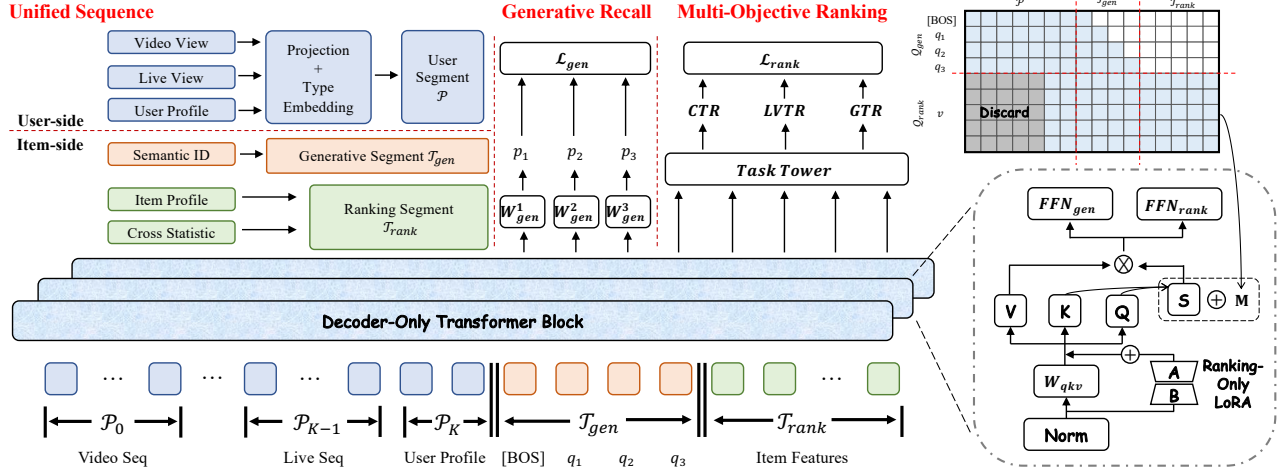


Figure 2: The overall architecture of our model.

Since $\mathcal{P}$ represents stable user side information and remains unchanged across different items, it serves as a shared memory read by each downstream target token, playing the role of an encoder key/value cache.

Generative segment $\mathcal{T}_{gen}$. Given the target item v with hierarchical SID $s_v = (q_1, q_2, q_3)$, this segment is the teacher-forced input of the generative recall task:

$$\mathcal{T}_{gen} = [\text{[BOS]}, e(q_1), e(q_2), e(q_3)], \quad (4)$$

where $e(\cdot)$ is the codebook embedding lookup and [BOS] is a learned special token. Position i of $\mathcal{T}_{gen}$ is trained to predict the next-level code q_i in the standard NTP pattern.

Ranking segment $\mathcal{T}_{rank}$. The SID alone is insufficient to fully model item characteristic or feature cross between (u, v) pair. Therefore, in addition to the SID of v , we append an ordered set of item feature tokens:

$$\mathcal{T}_{rank} = [e_1(v), e_2(v), \dots, e_M(v)], \quad (5)$$

which summarizes v from complementary views, e.g. item profile, content embedding, (u, v) cross statistics and upstream priors. Each $e_m(v)$ is produced by an independent lightweight projection over its raw feature block and shifted by a position type embedding. These tokens act as learnable ranking queries whose contextualized representations are used as sufficient feature fusion results to be transmitted to the multi-objective ranking tower.

Therefore, the complete input single sequence is:

$$S = [\underbrace{\mathcal{P}_0, \dots, \mathcal{P}_{K-1}, \mathcal{P}_K}_{\text{User segment}} \parallel \underbrace{[\text{[BOS]}, e(q_1), e(q_2), e(q_3)]}_{\text{Generative segment}} \parallel \underbrace{e_1(v), \dots, e_M(v)}_{\text{Ranking segment}}]. \quad (6)$$

This unified sequence composition design allows the recall and ranking targets to be fed into the same decoder-only Transformer block, where they sit within the same stack and share both model parameters and the contextual memory of invariant content.

4.2 Dual-Query Prefix-Causal Attention

A naïve practice for Transformer-based generative recall [28, 35] or ranking [22, 34] is to apply causal self-attention over the entire unified sequence. While conceptually straightforward, this design is computationally wasteful. First, the user segment is typically much longer than the two target segments, making it highly redundant to force the full sequence to perform attention at every layer. Second, the two target segments play roles, since $\mathcal{T}_{gen}$ follows an autoregressive tasks and therefore cannot access future positions, whereas $\mathcal{T}_{rank}$ consists of item-side information added synchronously and can thus be mutually visible. To address this, UniR² introduces a Dual-Query Prefix-Causal Attention mechanism (DQ-PCA), which keeps the two objectives separated along the sequence axis while sharing all base attention weights.

Dual queries with limited visibility. At each layer, for the two distinct target segments, the block consumes the unified sequence using two independent queries with different visibility patterns:

- The generative query Q^{gen} carries the SID tokens $\mathcal{T}_{gen}$. Consistent with standard generative recall, predicting possible SID only depends on user-side context. Therefore Q^{gen} attend over the full user segment and over its own preceding positions, implementing a prefix-causal pattern:

$$Q^{gen} = [\mathcal{T}_{gen}], \quad K^{gen} = V^{gen} = [\mathcal{P}, \mathcal{T}_{gen}]. \quad (7)$$

Formally, indexing SID query positions by $i \in [1, L]$ and key positions by $j \in [1, |\mathcal{P}| + L]$, the attention mask reads:

$$M_{i,j}^{gen} = \begin{cases} 0, & j \leq |\mathcal{P}| \text{ or } |\mathcal{P}| < j \leq |\mathcal{P}| + i, \\ -\infty, & |\mathcal{P}| + i < j \leq |\mathcal{P}| + L. \end{cases} \quad (8)$$

- The ranking query Q^{rank} carries the item-side feature tokens $\mathcal{T}_{rank}$. Since every element of the unified sequence is in principle mutually informative for scoring, bidirectional self-attention is the most expressive choice for feature fusion. However, considering that the long user history has already been amortized by the generative stage, and the sequential retrieval signals such as GSU [17] results are already encoded as tokens inside $\mathcal{T}_{rank}$

itself. Therefore Q^{rank} attend the partial sequence within the user segment is discarded while only the user profile is retained:

$$Q^{\text{rank}} = [\mathcal{T}_{\text{rank}}], \quad K^{\text{rank}} = V^{\text{rank}} = [\mathcal{P}_K, \mathcal{T}_{\text{gen}}, \mathcal{T}_{\text{rank}}]. \quad (9)$$

The additive mask is uniformly zero within constrained scope:

$$M_{i,j}^{\text{rank}} = 0, \quad \forall i \in [1, M], j \in [1, |\mathcal{P}_K| + L + M]. \quad (10)$$

It can be observed that the personalized identity information in the prefix user segment is read by both queries, which avoids redundant computation within a single forward pass. In addition, the ranking segment attends to the complete generative segment, so the ranking output is always produced from a representation that has already incorporated the recall trajectory corresponding to the target item. These make UniR² a genuinely unified model, rather than two co-located submodels sharing the same encoder, and it delivers representation coupling by construction.

Weights sharing. Similar to the normal decoder-only architecture, each block of UniR² consists of an attention mechanism and a Feed-Forward Network. The former is based on multi-head attention and uses DQ-PCA to construct corresponding inputs for different targets:

$$O = \text{MultiHead}(W_q Q, W_k K, W_v V) W_o, \quad (11)$$

The base projections $W_q, W_k, W_v, W_o \in \mathbb{R}^{d \times d}$ are shared between the two branches and across all memory partitions.

The outputs corresponding to different queries are then passed through task-specific feed-forward blocks FFN_{gen} and FFN_{rank} , thereby establishing a task-level barrier between the two objectives at each layer [5]. Stacking N layers of such blocks yields $\mathbf{h}^{\text{gen}} \in \mathbb{R}^{L \times d}$ for the recall head and $\mathbf{h}^{\text{rank}} \in \mathbb{R}^{M \times d}$ for the ranking head. In this way, the architecture fully preserves a shared backbone and the invariance of a single forward pass, which together serve as the foundation for unification.

4.3 Generative Recall

The generative branch aims at turning generative hidden states $\mathbf{h}^{\text{gen}}$ into a distribution over the hierarchical SID codebook. To let each SID position specialize to its own codebook level while still sharing the transformer trunk, we attach L independent output projections $W_{\text{gen}}^{(1)}, \dots, W_{\text{gen}}^{(L)} \in \mathbb{R}^{V \times d}$ on top of $\mathbf{h}^{\text{gen}}$ and predict the code of the next level i from the preceding hidden state:

$$p_{\Theta}(q_i | u, q_{<i}) = \text{softmax}(W_{\text{gen}}^{(i)} \mathbf{h}_{i-1}^{\text{gen}}). \quad (12)$$

The generation task is optimized based on a weighted cross-entropy:

$$\mathcal{L}_{\text{gen}}(\Theta) = \sum_{i=1}^L \alpha_i [-\log p_{\Theta}(q_i | u, q_{<i})], \quad (13)$$

with weights $\alpha_1 \geq \dots \geq \alpha_L > 0$ since the code placed ahead can determine the attributes of a generated item from a higher dimension. Because $\mathcal{L}_{\text{gen}}$ only reads $\mathbf{h}^{\text{gen}}$, its gradients flow exclusively through Q^{gen} 's attention paths and never reach the item-side components.

4.4 Multi-Objective Ranking

The ranking branch uses the contextualized ranking hidden states $\mathbf{h}^{\text{rank}}$ to estimate multiple business objectives for the candidate

items. Since the ranking model in recommendation expects multi-source complementary signals as input, $\mathbf{h}^{\text{rank}}$ only preserves target-aware deep interaction representations. Therefore, we additionally incorporate the original user and author features to prevent shallow signals from being smoothed out. Moreover, we also concatenate the output at the last SID position $\mathbf{h}_L^{\text{rank}}$. Although the ground-truth SID at this position is redundant for autoregressive computation in the generative task, this hidden state observes the entire generation trajectory and thus provides target semantics from a perspective different from the discriminative representation. Therefore, the final input to the ranker consists of:

$$z = \text{Concat}(u, v, \mathbf{h}_L^{\text{gen}}, \mathbf{h}^{\text{rank}}). \quad (14)$$

The vector z is fed into an MMoe [16] model, where each task-specific tower outputs $\hat{y}_t$ for objective t . The entire ranker is trained with the BCE loss:

$$\mathcal{L}_{\text{rank}}(\Theta) = \sum_{t=1}^T \omega_t \cdot \text{BCE}(\hat{y}_t, y_t), \quad (15)$$

where ω_t is the objective weight.

Gradient entanglement. In this way, UniR² completes the prediction of both generative recall and ranking within a single forward pass over one unified sequence. However, if we simply define the overall training objective as the sum of the two tasks:

$$\mathcal{L}(\Theta) = \mathcal{L}_{\text{gen}}(\Theta) + \mathcal{L}_{\text{rank}}(\Theta), \quad (16)$$

the gradients from both generative recall and ranking will be propagated into shared parameters, which can introduce cause optimization interference.

A straightforward solution is to apply stop-gradient before the ranking head: $\tilde{z} = \text{sg}(z)$. This blocks $\mathcal{L}_{\text{rank}}$ from back-propagating into the Transformer trunk and protects the generative path. However, it also freezes ranking-side feature fusion: the head can only learn on top of a representation mainly optimized for SID prediction. This often amplifies the seesaw effect, because stronger objectives dominate the head while weaker or delayed objectives cannot adapt the attention-level interactions. UniR² therefore keeps the stop-gradient boundary for safety, but adds a learnable low-rank path activated only for the ranking view.

LoRA-balanced ranking adaptation. We inject LoRA [9] into the attention projections used by the ranking branch. The base projections W_q, W_k, W_v remain shared with the generative branch, but their outputs are detached when they are consumed by the ranking query. Formally, ranking-specific adaptation is carried by low-rank residual matrices:

$$\Delta W_X^{\text{rank}} = B_X^{\text{rank}} A_X^{\text{rank}}, \quad X^{\text{rank}} = \text{sg}(WX) + \Delta W_X^{\text{rank}} X, \quad (17)$$

where $X \in \{Q, K, V\}$ is the inputs of the ranking branch, and $B_X^{\text{rank}} \in \mathbb{R}^{d \times r}, A_X^{\text{rank}} \in \mathbb{R}^{r \times d}$ are low-rank parameters. These modules take effect only in ranking-view attention in each layer. The generative query still uses the original base projections and never activates the ranking LoRA branch. Thus, LoRA is placed at the Q/K/V projections that determine ranking-side feature fusion, giving the ranker enough adaptive capacity while isolating the base generative backbone from ranking gradients.

In general, trainable parameters are divided into generative, ranking, and shared sparse embedding parts:

$$\Theta_{\text{gen}} \cap \Theta_{\text{rank}} = \emptyset, \quad \Theta = \Theta_{\text{gen}} \cup \Theta_{\text{rank}} \cup \Theta_{\text{emb}}. \quad (18)$$

Θ_{gen} contains the base Transformer projections, generative FFN, and recall heads; Θ_{rank} contains ranking LoRA matrices, ranking-side FFN parameters, and multi-task towers; Θ_{emb} contains sparse embeddings shared by both tasks [20]. UniR² therefore preserves representation coupling in the forward pass, while tasks separated in the optimization and modules.

5 Deployment

5.1 Training Strategy

Data sample. Mainstream generative recall methods essentially perform maximum-likelihood fitting over the online exposure distribution. Their training sets usually contain only exposed or positive feedback samples, without explicit discriminative negative samples. However, the ranker inside UniR² is jointly trained with the generative branch in the same forward pass. If we directly inherit the sampling strategy of generative recall, each XTR objective would lack negative contrast, leading to degraded scoring ability.

Therefore, we adopt full sample stream with target-adaptive masks strategy. All (u, v) pair samples entering the model are visible to the ranker, while each discriminative objective uses an independent sample mask and contributes loss only on the subset semantically relevant to that objective. This provides sufficient contrast for all objectives within the same batch. Above ranker stream, the generative branch further performs downsampling, which keeps all clicked samples and randomly exposed samples as positive samples for the GRM loss, so as to preserve the recall-side semantics of fitting the online exposure distribution. These two sampling strategies coexist within the same forward pass, and control whether to participate in the loss through their respective masks, thereby balancing the discriminative contrast required for ranking and the distribution fitting target required for recall.

Two-Stage Training. We adopt a two-stage joint training strategy. In the first stage, only the GRM loss in Eq.13 is enabled to obtain a backbone with stable distribution fitting. In the second stage, training is warm-started from this checkpoint, and the ranker loss is appended on the same sequence as Eq.16 for joint training.

This design is mainly motivated by the fact that the discriminative input of the ranker depends on the SID representations produced by the generative branch. If the ranker is introduced at the early stage when the SID semantics are still unstable and drifting, it can be misled by substantial noise and form incorrect decision boundaries, resulting in negative transfer. Therefore, the two-stage design decouples distribution fitting and discriminative alignment in the optimization space, imposing a curriculum prior on multi-task learning.

5.2 Inference Pipeline

In a conventional industrial recommender system, a user request is first processed by a recall model to retrieve hundreds of candidates from the entire item corpus. After candidate-level features are extracted, an independent ranking model evaluates each user-item pair and produces multi-objective scores. Between the recall and

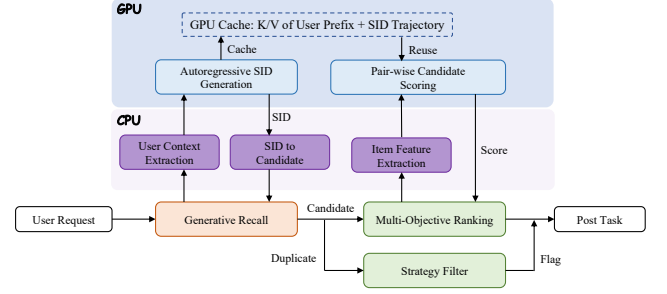


Figure 3: The inference pipeline of UniR². GPU caches the user context and SID-trajectory KV states computed during recall for direct reuse in ranking, while strategy filtering and ranking are executed in parallel.

ranking processes, the generated candidates should undergo further processing through commercial strategies such as qualification review, deduplication, and content filtering before being passed on to the downstream stages. This pipeline alternates between CPU and GPU execution. Feature extraction and service orchestration are mainly performed on CPUs, whereas the forward passes of the recall and ranking models are executed on GPUs. Because recall and ranking are deployed as independent services, the ranking stage needs to reconstruct the user context and perform another complete model forward pass, introducing redundant user-side computation, cross-service data transfer, and additional scheduling overhead.

UniR² replaces the cascaded recall and ranking models with a single serving instance. Given a user request, the service first extracts the user profile and historical behavior features and then performs autoregressive SID generation. During recall, the key and value representations of the user prefix, as well as the KV states corresponding to SID generation trajectory of each candidate v , are stored in the GPU KV cache:

$$\mathcal{C}_{u,v} = \{K_\ell(\mathcal{P}), V_\ell(\mathcal{P})\}_{\ell=1}^N \cup \{K_\ell(\mathcal{T}_{\text{gen}}^v), V_\ell(\mathcal{T}_{\text{gen}}^v)\}_{\ell=1}^N, \quad (19)$$

where N is the number of Transformer layers.

After recall, the CPUs extract the item-side attributes of the retrieved candidates. These attributes are projected into ranking tokens and appended to the cached unified sequence. The ranking query can therefore directly attend to the cached user context and the corresponding SID trajectory without recomputing either segment. The candidate ranking stage only needs to compute the newly appended item-feature tokens and ranking-specific LoRA paths. Moreover, due to the full view, there is no need for autoregression and only a single sequence calculation is required.

A direct implementation could sequentially execute recall, strategy filtering, and ranking. However, in our production scenario, the external strategy service takes longer than the UniR² ranking forward pass, which makes sequential execution offset the latency benefit of model unification. To avoid this redundant time, we duplicate the recalled candidate set and dispatch it to two parallel branches. The strategy branch transfers the candidate attributes to the filtering service and produces a binary filter flag for each candidate. Meanwhile, the ranking branch extracts item-side features and computes multi-objective scores for all candidates in a

Table 1: The overall recall performance comparison of different models on live-streaming offline dataset. The best results are boldfaced and the second-best results are underlined, with the relative improvements denoted as *Imprv.*↑.

Models	Show				Click			
	HR@64	MRR@64	HR@128	MRR@128	HR@64	MRR@64	HR@128	MRR@128
KuaiFormer	0.4176	0.1328	0.4907	0.1337	0.4479	0.1499	0.5164	0.1507
GNN	0.4552	0.1532	0.5269	0.1540	0.5463	0.1941	0.6197	0.1949
OneRec	0.6844	0.3224	0.7261	0.3230	0.7494	0.3987	0.7789	0.3992
OneLive	0.7177	0.3254	0.7742	0.3260	0.7799	0.4046	0.8284	0.4051
NEZHA	0.7244	0.3221	0.7794	0.3228	0.7848	0.4062	0.8312	0.4067
PROMISE	<u>0.7329</u>	<u>0.3391</u>	<u>0.7874</u>	<u>0.3396</u>	<u>0.7932</u>	<u>0.4286</u>	<u>0.8416</u>	<u>0.4289</u>
UniR ²	0.7662	0.3506	0.8114	0.3563	0.8309	0.4441	0.8741	0.4516
<i>Imprv.</i> ↑	+4.54%	+3.39%	+3.05%	+4.92%	+4.75%	+3.62%	+3.86%	+5.29%

Table 2: The overall rank performance comparison of UniR² with baseline on live-streaming offline dataset.

Models	CTR		LVTR		GTR	
	AUC	UAUC	AUC	UAUC	AUC	UAUC
Base	0.8450	0.6635	0.8994	0.7591	0.9541	0.6589
UniR ²	0.8513	0.6731	0.9096	0.7623	0.9556	0.6601
<i>Imprv.</i> ↑	+0.75%	+1.45%	+1.13%	+0.42%	+0.16%	+0.18%

batched GPU forward pass. Once ranking is completed, it waits only for the filter flags and removes candidates that violate the strategy constraints, hiding ranking computation behind the strategy latency. This single-service deployment simultaneously eliminates repeated user-context computation, preserves the recall trajectory for ranking, and overlaps model inference with external strategy execution. The pipeline is illustrated in Figure 3.

6 Experiments

In this section, we conduct extensive experiments offline and online to answer the following questions:

- **RQ1:** How does our unified model perform compared to the recall and ranking baselines?
- **RQ2:** How does the proposed key structural components contribute to the overall model performance?
- **RQ3:** How does our unified model perform in online services?

6.1 Experimental Settings

6.1.1 Dataset. We conduct our evaluation based on large-scale industrial live-streaming platform logs from Kuaishou App. This dataset includes 400 million users and 3 million authors, as well as billions of diverse interactions among them. For generative recall task, we mainly use the exposure (show) and click records for test. For multi-objective ranking task, we mainly report the core objectives such as click, long-view and gift, with the corresponding through rate denoted as CTR, LVTR and GTR.

6.1.2 Evaluation Metrics. The evaluation considers both recall and ranking performance, therefore we adopt Accuracy (ACC), Hit

Rate (HR) and Mean Reciprocal Rank (MRR) for recall task and AUC, UAUC for ranking task, which are all widely used offline evaluation metrics in prior works [4, 17, 26, 34].

6.1.3 Baselines. For the recall task, we compare UniR² with two groups of baselines: (i) traditional recall methods, including the long-sequence model KuaiFormer [13] and graph-structured model GNN [8]; (ii) generative methods, including OneRec [4] and OneLive [26], as well as the efficient decoding framework NEZHA [28] and the process reward model PROMISE [7]. The latter two baselines and UniR² are all modified based on OneLive. For the ranking task, our baseline is an MMoE-style multi-objective ranking model HoME [27] with conventional feature fusion, which has been deployed online and serves as the basic of our platform. We directly compare our unified model with the online production models.

6.2 Overall Experiments (RQ1)

6.2.1 Recall Performance. As the results shown in Table 1, UniR² consistently outperforms both conventional and generative baselines in terms of generative recall, demonstrating that the unified sequence formulation does not sacrifice the model’s core capability of generating SIDs. This improvement can be attributed to two key factors. First, our carefully designed attention mechanism enables SIDs in the recall branch to effectively perceive the rich user’s interest representations encoded from historical interactions through prefix-cross attention. Second, the dual branches design decouples gradients for model parameters, thereby mitigating the conflicting influence of the ranking objective on generative capability. Meanwhile, the retained shared embeddings still allow the shallow signals to benefit from fine-grained optimization through positive-negative sample contrast, which may further explain the substantial improvement in generative recall performance.

6.2.2 Rank Performance. We compare UniR² with the online ranking model in Table 2. As can be observed, UniR² improves XTR prediction performance on both sparse and dense targets, demonstrating that the shared decoder-only sequence architecture can serve as an effective backbone for feature fusion. The ranking module does not merely passively reuse the generative backbone. Instead, it preserves task-specific feature interactions for ranking through LoRA-based adaptation. Besides, the unified design benefits

Table 3: The overall performance comparison of variants with different components ablation.

Branch	Variants	Recall			Ranking			Efficiency	
		$\mathcal{L}_{gen} \downarrow$	HR@1 $\uparrow$	ACC@all $\uparrow$	$\mathcal{L}_{rank} \downarrow$	AUC@CTR $\uparrow$	AUC@GTR $\uparrow$	Params	FLOPs
	UniR ²	6.8260	0.1826	0.6249	5.9528	0.8450	0.9541	75M	9.37G
Recall	w. lazy decoder-only	+0.92%	-1.92%	-0.46%	-	-	-	74M	7.42G
	w. causal self-attention	-0.89%	+2.85%	+0.46%	-	-	-	75M	45.09G
Ranking	w/o. DQ-PCA & sg	+30.10%	-44.39%	-10.57%	-3.83%	+3.97%	+5.82%	72M	7.43G
	w/o. history discard	+0.01%	+0.00%	-0.01%	-0.34%	+2.38%	+2.87%	75M	15.39G
	w/o. LoRA	-0.02%	+0.01%	+0.04%	+7.05%	-13.79%	-31.29%	75M	9.27G

from the generated SID trajectory as an additional target-aware semantic view, complementing the original ranking features while preserving the discriminative contrast required by ranking.

Overall, the offline results verify the effectiveness of UniR² on both generative recall and multi-objective ranking. It achieves stronger candidate retrieval quality while effectively improves XTR prediction performance. This indicates that a shared decoder-only sequence architecture can integrate distribution fitting of generation and contrast drive of ranking within a single model.

6.3 In-depth Analysis (RQ2)

6.3.1 Components Ablation. We conducted extensive experiments on the core components of UniR² to quantify the contributions of these designs. The results are presented in Table 3. We summarize the observations as follows:

- For the generative recall branch, since the lazy decoder-only architecture in the baseline [26, 35] decouples user history encoding from SID generation, which leads to fragmented learning and representation sharing. Applying full-sequence causal self-attention brings only limited improvement while introducing substantially higher computational cost. In contrast, UniR² achieves a more favorable effectiveness and efficiency trade-off through prefix-cross attention, which provides sufficient access to historical interests without densely encoding the entire user history sequence at every layer.
- For the ranking branch, removing DQ-PCA and corresponding gradient truncation causes the model to strongly tilt toward ranking after long joint training, where ranking metrics improve but generative performance drops significantly. This phenomenon comes from the optimization asymmetry between the two tasks. Ranking is supervised by multiple dense discriminative objectives and can provide more frequent gradients, while SID generation is a structured autoregressive task that requires preserving token-level semantic consistency. Without dual-query visibility control and stop-gradient isolation, ranking gradients can directly reshape the shared attention representations, making the backbone increasingly optimized for discriminative scoring rather than stable SID generation.
- Discarding the long history sequence for the ranking query greatly reduces FLOPs while causing little degradation in ranking performance. This suggests that the ranking branch does not need to repeatedly attend to the full user history. The generated SID trajectory has already condensed the user-item matching process

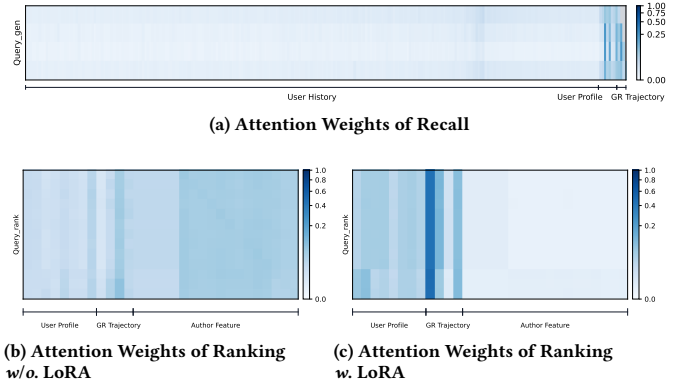


Figure 4: Visualization of attention weights of recall and ranking at an intermediate layer.

from the recall branch, and item-side sequential features such as GSU signals further summarize similar interests. Therefore, attending to the SID trajectory together with compact ranking features is sufficient for feature fusion.

- The LoRA is designed to provide ranking-specific adaptation capacity under optimization isolation. Gradient truncation protects the generative backbone but freeze the attention-level feature interactions available to the ranker. By adding low-rank residual paths only to the ranking-view attention projections, UniR² allows the ranker to learn task-specific feature fusion without modifying the base generative parameters and introducing only a small amount of parameters and computational load. Its impact on sparse targets such as gift is particularly significant.

6.3.2 Attention Analysis. We visualize the attention weights corresponding to the recall and ranking queries in the unified sequence in Figure 4. For generative recall, the SID query attends broadly to the behavioral prefix. This pattern is consistent with the hierarchical generation process, while the historical behaviors provide fine-grained evidence of dynamic interests, the profile features offer stable personalized priors, and previously generated SID tokens constrain the semantics of the next code.

The ranking-side comparison further reveals the role of LoRA-based adaptation. Without LoRA, attention inherited from the recall-oriented base projections is relatively diffuse and places substantial mass on author features. After enabling ranking-specific LoRA, the

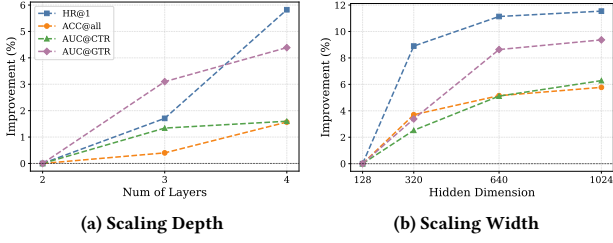


Figure 5: Performance improvements obtained by scaling the depth and width of the unified architecture.

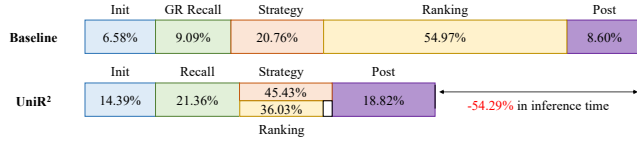


Figure 6: The comparison of system inference time.

distribution becomes much sharper around the user profile and generation trajectory. The ranker therefore learns to emphasize target-aware user-item matching signals that have been distilled by the recall process, rather than merely reusing a generic representation learned for SID prediction. This verifies that the generation trajectory serves as an effective representation bridge and that LoRA restores ranking-specific feature fusion without perturbing the generative backbone.

6.3.3 Scaling Unified Architecture. A key intuition behind the unified architecture is that both recall and ranking objectives can benefit from scaling the shared backbone. We therefore conduct depth and width scaling experiments, with the results shown in Figure 5. As can be observed, scaling from both brings substantial improvements in generative and ranking capabilities, which verifies the strong scalability of UniR². Notably, a larger unified backbone enhances both distribution fitting and discriminative prediction without inducing an imbalanced trade-off between them. This suggests that the optimization isolation mechanism may prevent the additional shared capacity from being monopolized by a single task. However, the improvements also exhibit clear diminishing returns. Therefore, we ultimately choose the layer of 3 and the hidden dimension of 640 for the online service to balance model performance and resource consumption.

6.3.4 Inference Latency. We compare the end-to-end latency breakdown between the online pipeline and UniR², with the results shown in Figure 6. In the baseline, recall and ranking are executed by two independent models, with the ranking stage dominating the overall inference time. Together with repeated initialization, feature preparation, and information transfer across pipeline, this leads to redundant time consumption. UniR² instead reuses the same user context and decoder-only architecture for both tasks, and appends the feature fusion required for ranking after the generated SID trajectory. As a result, reusing cached representations substantially

reduces the overall pipeline latency. In addition, our carefully designed parallelization strategy and ranking scheme further reduce end-to-end inference time. This result shows that unification provides not only representation and objective consistency, but also a concrete serving advantage by eliminating redundant computation and hiding ranking latency within the existing pipeline schedule.

6.4 Online A/B Test (RQ3)

To evaluate the real-world business impact of UniR², we deploy it to the live-streaming services of both Kuaishou and Kuaishou Lite App. UniR² simultaneously replaces the original OneLive-based recall model and the deployed production pre-ranking model, which together serve as the baseline assembled in cascaded architecture. The online A/B test was launched in June 2026, lasted for two weeks, and exposed to 5% of online traffic.

As shown by the online results, UniR² brings consistent gains on both platforms. On Kuaishou App, it improves play volume by +1.177%, follow rate by +0.655% and like rate by +2.560%. On Kuaishou Lite App, it improves the number of gifting users by +0.717%, gifting intention by +1.567%, and total gifting amount by +2.569%. The positive improvements across multiple behavior metrics demonstrate that UniR² can both retrieve suitable candidates and produce high-quality user-item pair scores within a single model. These online gains further validate the practicality and effectiveness of unified recall and ranking in large-scale industrial recommendation systems.

7 Conclusion

In this work, we propose UniR², a unified decoder-only Transformer that organizes user context, SID trajectory, and item features within a single heterogeneous sequence. Its Dual-Query Prefix-Causal Attention enables causal SID generation and expressive ranking feature fusion under task-specific visibility, while task-specific forward paths and ranking-side LoRA balance forward representation sharing with optimization isolation. Extensive offline experiments demonstrate consistent improvements in both generative recall and multi-objective ranking, and further verify that the SID trajectory serves as an effective representation bridge between the two tasks. By reusing cached user-context and SID states, UniR² also reduces end-to-end inference time. Long-term online A/B tests show consistent positive gains, validating its effectiveness and practicality in large-scale industrial recommendation systems.

References

- [1] Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. 2025. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. 247–256.
- [2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
- [3] Sunhao Dai, Jiakai Tang, Jiahua Wu, Kun Wang, Yuxuan Zhu, Bingjun Chen, Bangyang Hong, Yu Zhao, Cong Fu, Kangle Wu, et al. 2025. Onepiece: Bringing context engineering and reasoning to industrial cascade ranking system. *arXiv preprint arXiv:2509.18091* (2025).
- [4] 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. *arXiv preprint arXiv:2502.18965* (2025).
- [5] Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. 2026. Sensenova-u1:

- Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500* (2026).
- [6] Kairui Fu, Changfa Wu, Kun Yuan, Binbin Cao, Dunxian Huang, Yuliang Yan, Junjun Zheng, Jianing Zhang, Silu Zhou, Jian Wu, et al. 2026. RankGR: Rank-Enhanced Generative Retrieval with Listwise Direct Preference Optimization in Recommendation. *arXiv preprint arXiv:2602.08575* (2026).
 - [7] Chengcheng Guo, Kuo Cai, Yu Zhou, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, and Guorui Zhou. 2026. PROMISE: Process Reward Models Unlock Test-Time Scaling Laws in Generative Recommendations. *arXiv preprint arXiv:2601.04674* (2026).
 - [8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
 - [9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.
 - [10] Xu Huang, Hao Zhang, Zhifang Fan, Yunwen Huang, Zhuoxing Wei, Zheng Chai, Jinan Ni, Yuchao Zheng, and Qiwei Chen. 2026. MixFormer: Co-Scaling Up Dense and Sequence in Industrial Recommenders. *arXiv preprint arXiv:2602.14110* (2026).
 - [11] Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. 2026. Spend Search Where It Pays: Value-Guided Structured Sampling and Optimization for Generative Recommendation. *arXiv preprint arXiv:2602.10699* (2026).
 - [12] Hanyu Li, Yi-Ping Hsu, Aditya Mantha, Prabhat Agarwal, Laksh Bhasin, Jialu Wang, Hongtao Lin, Bella Huang, Yaxin Li, Xinyi Li, et al. 2026. UniPinRec: Unifying Generative Retrieval and Ranking at Pinterest Scale. *arXiv preprint arXiv:2606.00422* (2026).
 - [13] Chi Liu, Jiangxia Cao, Rui Huang, Kai Zheng, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. Kuaiformer: Transformer-based retrieval at kuaishou. *arXiv preprint arXiv:2411.10057* (2024).
 - [14] Yu Liu and Jiangxia Cao. 2026. Harmonizing Generative Retrieval and Ranking in Chain-of-Recommendation. *arXiv preprint arXiv:2604.25787* (2026).
 - [15] Xinchun Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2025. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 5915–5922.
 - [16] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.
 - [17] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.
 - [18] 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.
 - [19] Dekai Sun, Yiming Liu, Jiafan Zhou, Xun Liu, Chenchen Yu, Yi Li, Jun Zhang, Huan Yu, and Jie Jiang. 2026. OneRanker: Unified Generation and Ranking with One Model in Industrial Advertising Recommendation. *arXiv preprint arXiv:2603.02999* (2026).
 - [20] Yijia Sun, Shanshan Huang, Zhiyuan Guan, Qiang Luo, Ruiming Tang, Kun Gai, and Guorui Zhou. 2026. GRank: Towards Target-Aware and Streamlined Industrial Retrieval with a Generate-Rank Framework. In *Proceedings of the ACM Web Conference 2026*. 7798–7808.
 - [21] Hongyan Tang, Junling Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM conference on recommender systems*. 269–278.
 - [22] Jiakai Tang, Runfeng Zhang, Weiqiu Wang, Yifei Liu, Chuan Wang, Xu Chen, Yeqiu Yang, Jian Wu, Yuning Jiang, and Bo Zheng. 2026. LoopCTR: Unlocking the Loop Scaling Power for Click-Through Rate Prediction. *arXiv preprint arXiv:2604.19550* (2026).
 - [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
 - [24] Chunqi Wang, Bingchao Wu, Taotian Pang, Jiahao Wang, Jie Yang, Jia Liu, Hao Zhang, Hai Zhu, Lei Shen, Shizhun Wang, et al. 2026. SORT: A Systematically Optimized Ranking Transformer for Industrial-scale Recommenders. *arXiv preprint arXiv:2603.03988* (2026).
 - [25] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.
 - [26] Shen Wang, Yusheng Huang, Ruochen Yang, Shuang Wen, Pengbo Xu, Jiangxia Cao, Yueyang Liu, Kuo Cai, Chengcheng Guo, Shiyao Wang, et al. 2026. OneLive: Dynamically Unified Generative Framework for Live-Streaming Recommendation. *arXiv preprint arXiv:2602.08612* (2026).
 - [27] Xu Wang, Jiangxia Cao, Zhiyi Fu, Kun Gai, and Guorui Zhou. 2025. Home: Hierarchy of multi-gate experts for multi-task learning at kuaishou. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 2638–2647.
 - [28] Yejing Wang, Shengyu Zhou, Jinyu Lu, Ziwei Liu, Langming Liu, Maolin Wang, Wenlin Zhang, Feng Li, Wenbo Su, Pengjie Wang, et al. 2026. NeZha: A zero-sacrifice and hyperspeed decoding architecture for generative recommendations. In *Proceedings of the ACM Web Conference 2026*. 8073–8082.
 - [29] Zhipeng Wei, Kuo Cai, Junda She, Jie Chen, Minghao Chen, Yang Zeng, Qiang Luo, Wencong Zeng, Ruiming Tang, Kun Gai, et al. 2026. Oneloc: Geo-aware generative recommender systems for local life service. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*. 735–744.
 - [30] Ruochen Yang, Yueyang Liu, Zijie Zhuang, Changxin Lao, Yuhui Zhang, Jiangxia Cao, Jia Xu, Xiang Chen, Haoke Xiao, Xiangyu Wu, et al. 2026. SARM: LLM-Augmented Semantic Anchor for End-to-End Live-Streaming Ranking. *arXiv preprint arXiv:2602.09401* (2026).
 - [31] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.
 - [32] Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Daifeng Guo, Yanli Zhao, Shen Li, Yuchen Hao, Yantao Yao, et al. 2024. Wukong: Towards a scaling law for large-scale recommendation. *arXiv preprint arXiv:2403.02545* (2024).
 - [33] Kun Zhang, Jingming Zhang, Wei Cheng, Yansong Cheng, Jiaqi Zhang, Hao Lu, Xu Zhang, Haixiang Gan, Jiangxia Cao, Tenglong Wang, et al. 2026. OneMall: One Model, More Scenarios—End-to-End Generative Recommender Family at Kuaishou E-Commerce. *arXiv preprint arXiv:2601.21770* (2026).
 - [34] Zhaoqi Zhang, Haolei Pei, Jun Guo, Tianyu Wang, Yufei Feng, Hui Sun, Shaowei Liu, and Aixin Sun. 2026. Onetrans: Unified feature interaction and sequence modeling with one transformer in industrial recommender. In *Proceedings of the ACM Web Conference 2026*. 8162–8170.
 - [35] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, et al. 2025. Onerec-v2 technical report. *arXiv preprint arXiv:2508.20900* (2025).
 - [36] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.
 - [37] Yifeng Zhou, Yuehong Hu, Zhixiang Feng, Junwei Pan, Kaihui Wu, Hanyong Li, Shangyu Zhang, Shudong Huang, Zhangbin Zhu, Chengguo Yin, et al. 2026. TokenFormer: Unify the Multi-Field and Sequential Recommendation Worlds. *arXiv preprint arXiv:2604.13737* (2026).