One Model to Rank Them All: Unifying Online Advertising with End-to-End Learning

Junyan Qiu*†
Meituan
Shanghai, China
qiujunyan@meituan.com

Ze Wang*†
Meituan
Shanghai, China
wangze18@meituan.com

Fan Zhang
Meituan
Shanghai, China
zhangfan133@meituan.com

Zuowu Zheng
Meituan
Shanghai, China
zhengzuowu@meituan.com

Jile Zhu
Meituan
Shanghai, China
zhujile@meituan.com

Jiangke Fan
Meituan
Shanghai, China
jiangke.fan@meituan.com

Teng Zhang
Meituan
Shanghai, China
zhangteng09@meituan.com

Haitao Wang
Meituan
Chengdu, China
wanghaitao13@meituan.com

Xingxing Wang
Meituan
Beijing, China
wangxingxing04@meituan.com

Abstract

Modern industrial advertising systems commonly employ Multi-stage Cascading Architectures (MCA) to balance computational efficiency with ranking accuracy. However, this approach presents two fundamental challenges: (1) performance inconsistencies arising from divergent optimization targets and capability differences between stages, and (2) failure to account for advertisement externalities - the complex interactions between candidate ads during ranking. These limitations ultimately compromise system effectiveness and reduce platform profitability. In this paper, we present **UniROM**, an end-to-end generative architecture that **Uni**fies online advertising **R**anking as **O**ne **M**odel. UniROM replaces cascaded stages with a single model to directly generate optimal ad sequences from the full candidate ad corpus in location-based services (LBS). The primary challenges associated with this approach stem from high costs of feature processing and computational bottlenecks in modeling externalities of large-scale candidate pools. To address these challenges, UniROM introduces an algorithm and engine co-designed hybrid feature service to decouple user and ad feature processing, reducing latency while preserving expressiveness. To efficiently extract intra- and cross-sequence mutual information, we propose RecFormer with an innovative cluster-attention mechanism as its core architectural component. Furthermore, we propose a bi-stage training strategy that integrates pre-training with reinforcement learning-based post-training to meet sophisticated platform and advertising objectives. Extensive offline evaluations on public benchmarks and large-scale online A/B testing on industrial advertising platform have demonstrated the superior performance of UniROM over state-of-the-art MCAs.

CCS Concepts

• **Information systems** → **Computational advertising**; **Display advertising**; • **Computing methodologies** → **Neural networks**.

Keywords

Online Advertising, Recommender System, End-to-End Architecture, Non-autoregressive Generation.

ACM Reference Format:
Junyan Qiu, Ze Wang, Fan Zhang, Zuowu Zheng, Jile Zhu, Jiangke Fan, Teng Zhang, Haitao Wang, and Xingxing Wang. 2025. One Model to Rank Them All: Unifying Online Advertising with End-to-End Learning. In . ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online advertising serves as a cost-efficient and precise channel for advertisers to promote contents to millions of online users, which has become the main revenue source for many platforms [4, 9, 15]. To balance computational efficiency and prediction accuracy, most industrial advertising systems widely adopt the Multi-stage Cascading Architectures (MCAs) [1, 12, 26, 27]. As illustrated in Figure 1, a typical MCA decomposes ad ranking problem into four sequential stages: recall, pre-ranking, ranking, and auction. Each stage progressively filters the candidate pool by selecting top-performing ads from its input list before passing them to the subsequent stage.

Although the MCA paradigm has proven efficacy, it is constrained by inconsistencies across stages, which stem from divergent modeling objectives and uneven capacity distribution. These limitations hinder the holistic optimization of platform-wide objectives. For instance, lightweight recall models prioritize speed

*Equal contribution.
†Corresponding authors.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

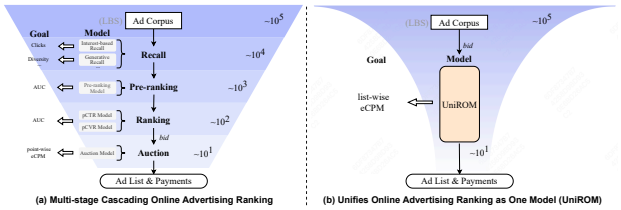


Figure 1: Illustrations of multi-stage cascading architecture and unified online advertising ranking.

over accuracy, while subsequent ranking stages employ complex architectures for precise CTR estimation. This misalignment creates prediction discrepancies that propagate through the pipeline, ultimately degrading the quality of final displayed ads [28]. Second, ignored externalities—the mutual influence among candidate ads—fundamentally limit performance. Most existing ranking approaches predominantly rely on the independent CTR assumption [13], failing to account for how ad permutations or contextual interactions shape user preferences.

To address optimization conflicts in MCAs, the development of coordinated learning frameworks that reconcile inter-stage objectives represents a critical research frontier. Some approaches employ a joint consistency loss computation between the pre-ranking and ranking stages to achieve rank alignment [5, 31, 33]. However, pre-ranking still deals with a much larger pool of candidates than ranking. Relying solely on ranking logs fails to adapt to changes in the retrieval distribution, resulting in a sample selection bias problem [32]. Recent advances in multistage optimization have introduced novel paradigm shifts through dynamic sample construction strategies, where each stage operates on specially designed training sets that maintain task-specific characteristics while preserving inter-stage dependency [22, 32, 35].

Despite these efforts to enhance overall recommendation performance by enabling interaction among rankers, existing approaches predominantly treat each ranker independently, thereby preserving a fragmented architecture where complementary strengths among models remain underutilized. Recent breakthroughs in large language models (LLMs) have catalyzed transformative developments in recommendation systems [18, 23, 29, 30], enabling the direct generation of personalized item sequences from user interaction histories. By framing recommendation as sequential transduction tasks, HSTU [30] scales to trillions of parameters and delivers outstanding performance. But it risks vulnerability to cold-start issues and embedding instability due to the dynamic, high-cardinality nature of item IDs. Recently, there emerges a new line of research that indexes items with meaningful IDs using vector quantization algorithms and generate items from the entire item set for recommendation [2, 18, 29, 34]. Although these methods have the potential to unify cascaded stages into a single model, the inherent sequential dependency in auto-regressive paradigm prohibits parallel computation and results in suboptimal inference latency in online advertising system.

In this paper, we present **UniROM**, a novel framework that **Uni**fies online advertising **R**anking as **O**ne **M**odel in location-based services (LBS). By narrowing the candidate set to ads within the same city (reducing its size to $\sim 10^5$), UniROM can perform fine-grained contextual modeling across the entire candidate space, enabling both more accurate item representations and sufficient exploration of user interests. Furthermore, this architecture conceptualizes online advertising systems as a unified generative process, thereby eliminating the inherent goal conflicts between different pipeline stages. Nevertheless, the proposed architecture is confronted with three fundamental challenges:

- **Costly feature processing.** The reliance on complex feature services in online advertising leads to significant storage and transmission overhead, especially for cross-feature interactions between massive user and ad inventories.
- **Computational intensity of intra-sequence modeling** Effective advertising requires modeling deep user interests and contextual interactions, but advanced techniques like multi-head attention [25] face scalability issues due to quadratic computational complexity, making them impractical for large-scale online advertising systems.
- **Misalignment with advertising objectives.** Existing generative approaches, primarily developed for recommendations, fail to address critical advertising-specific requirements including auction dynamics, bid optimization, and advertiser objectives.

To minimize feature storage and transmission overhead while maintaining expressiveness, we develop a novel Hybrid Feature Service (HFS) that optimizes computational efficiency by decoupling user and ad features. Additionally, by enabling batch processing of multiple candidates in a single pass, HFS significantly amortizes computational and I/O costs, thereby improving system scalability. Furthermore, we present RecFormer, an transformative recommendation framework that leverages an innovative cluster-attention mechanism to simultaneously model users' deep interests and contextual externalities while maintaining computational efficiency. To substantially enhance inference speed without compromising key platform optimization objectives, such as user engagement and revenue performance, UniROM incorporates an AucFormer module that generates advertising sequences through non-autoregressive processing. Finally, we explore a multi-stage training strategy to ensure consistency with platform objectives and auction constraints. To summarize, our main contributions are in three-fold:

- We present an innovative End-to-End Generative architecture capable of directly producing ad sequences through a unified model framework. To the best of our knowledge, this is the first industrial-grade solution that comprehensively addresses these challenges.
- We propose a novel algorithm-engineer co-design framework that integrates hybrid feature service with cluster-attention based modules, facilitating the evolution of advertising systems from feature-centric updates to computation-driven scalability.
- These contributions are rigorously validated through offline experiments on industrial datasets and large-scale A/B tests, demonstrating statistically significant improvements in CTR (+5.2%), RPM (+13.6%), and advertiser ROI (+3.1%) over state-of-the-art MCAs.

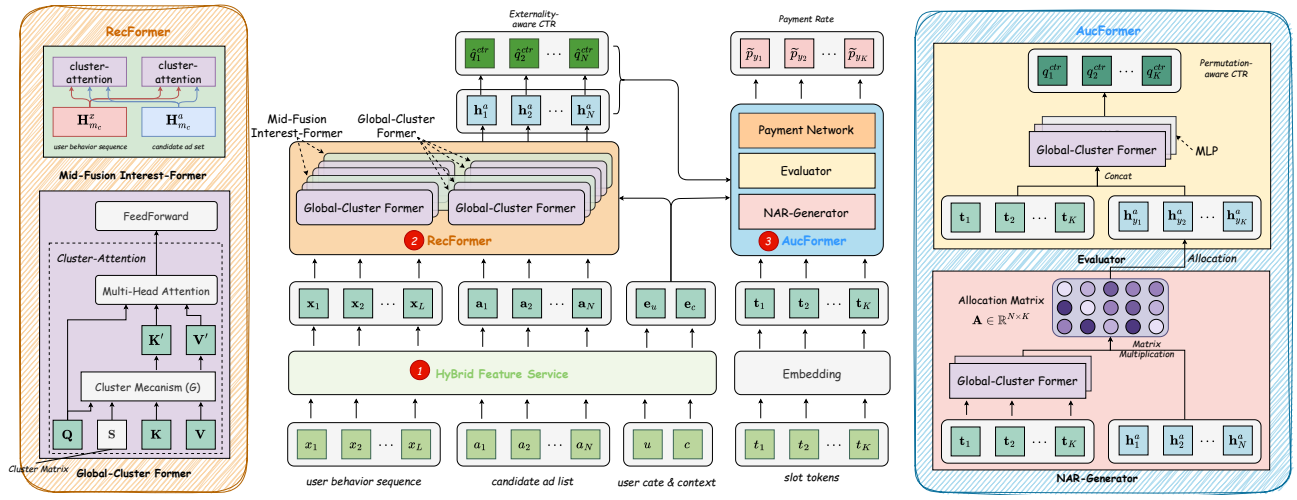


Figure 2: Architecture Overview of UniROM, showcasing its key components including the Hybrid Feature Service, RecFormer and AucFormer.

2 RELATED WORKS

Multi-stage Cascading Architectures (MCAs) have become the de facto paradigm in industrial advertising systems to balance computational efficiency and prediction accuracy [16, 22]. A typical implementation comprises four stages: recall, pre-ranking, ranking, and auction. The recall stage employs lightweight methods like collaborative filtering [7] and SASRec [10] to efficiently retrieve tens of thousands of candidates from the ad corpus. While these approaches enable low-latency computation through simplified architectures, their restricted model capacity inherently limits both feature representation power and ultimate performance ceiling. In the pre-ranking stage, models such as DSSM [8] and COLD [28] attempt to approximate ranking model performances under stricter latency constraints. However, the model capacity disparities often leads to prediction discrepancies between pre-ranking and subsequent ranking stages, ultimately compromising system consistency. The ranking stage utilizes sophisticated architectures like DIN [36] and SIM [21] with enriched features for precise CTR estimation. Nevertheless, most existing approaches operate under the separable CTR assumption [13], failing to account for mutual influences among candidate ads (i.e., externalities) that significantly affect user preferences in real-world scenarios. The final auction stage implements mechanisms including GSP [4], DNA [17], and CGA [37] to allocate ads based on platform revenue objectives. However, the cascaded nature of MCA restricts these mechanisms' capacity to model complete externalities, as early-stage filtering substantially reduces the candidate space before auction occurs.

3 METHODOLOGY

As illustrated in Figure 2, UniROM operates through three integrated components: 1) The Hybrid Feature Service (HFS) efficiently extracting and processing large-scale, fine-grained features; 2) RecFormer modeling deep user interests and candidate externalities across the full candidate set; and 3) AucFormer optimizing ranking to align with advertising platform objectives. Finally, we employ a bi-stage training strategy, wherein the pre-training phase focuses on modeling user preferences, while the post-training phase optimizes for profitability.

3.1 Preliminaries

Basic task. We describe a typical task in online advertising systems. Formally, when a PV (page view) request from the user u arrives, there are N advertisers competing for K ad slots ($K < N$), denoted as $\mathcal{C} = \{a_1, a_2, \ldots, a_N\}$. Each advertiser i submits a click bid b_i based on its private click value v_i. And ad systems output the pCTR denoting the probability that the user clicks the ad. Given ad auction mechanism $\mathcal{M}\langle\mathcal{A}, \mathcal{P}\rangle$, the goal is to propose a winning ad sequence $Y = (a_{y_i} \mid a_{y_i} \in \mathcal{C}, \forall i \in [K])$ that maximizes the expected revenue of ad platform, as follows:

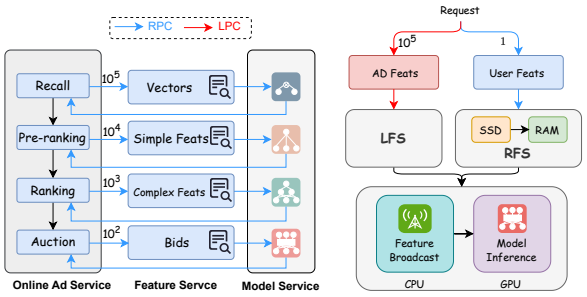
$$\underset{\theta}{\texttt{maximize}} \quad \mathbb{E}_Y\Big(\sum_{i=1}^{K} p_i \times pCTR_i\Big), \tag{1}$$

where θ is parameter of ad systems and p_i represents the payment of the i-th advertiser in ad sequence Y.

Auction constraints. Unlike recommender systems, ad systems not only maximize platform revenue but also ensure the utility for advertisers. Given the auction mechanism $\mathcal{M}\langle\mathcal{A}, \mathcal{P}\rangle$, an advertiser's expected utility u_i can be expressed as:

$$u_i(v_i; \boldsymbol{b}) = (v_i - p_i) \times pCTR_i. \tag{2}$$

For the design of ad auction mechanisms, two essential properties of ad auction: dominant strategy incentive compatible (DSIC, or IC) and individually rational (IR) are standard economic constraints that must be considered [11]. Specifically, an auction mechanism $\mathcal{M}\langle\mathcal{R}, \mathcal{P}\rangle$ is IC, if for each advertiser truthfully reports his bid $b_i = v_i$ and then his utility is maximized. Formally, let $\boldsymbol{b} = \{b_1, b_2, \ldots, b_N\}$ be the bid profile of all ads, we use $\boldsymbol{b}_{-i}$ to represent



(a) Remote Feature Service (b) Hybrid Feature Service

Figure 3: Architectures of two feature services.

the bid profile of all ads except ad i. For each ad i, it holds that

$$u_i(v_i; v_i, \boldsymbol{b}_{-i}) \geq u_i(v_i; b_i, \boldsymbol{b}_{-i}), \forall v_i, b_i \in \mathbb{R}^+, \tag{3}$$

and an auction mechanism is IR, if for each advertiser would not be charged more than his bid, as follows:

$$p_i \leq b_i, \forall i \in [N]. \tag{4}$$

Problem formulation. The complete problem can be described as developing an architectural solution to successfully select a winning ad sequence with maximum platform revenue Y from a wide range of candidates X, while adhering to the constraints of IC and IR. The objective is formulated as follows:

3.2 Hybrid Feature Service

Traditional Multi-stage Cascading Architectures (MCAs) rely on Remote Feature Services (RFS) to manage high-dimensional feature spaces involving categorical, numerical, cross-modal, and sequential interactions [21, 30]. As illustrated in Figure 3 (a), RFS extracts features from distributed storage and transmits them via Remote Procedure Calls (RPC) across stages. While effective for multi-stage coordination, this approach introduces two critical limitations. First, the explicit cross-feature space between hundreds of millions of users and tens of millions of candidate ads is too large, creating prohibitive storage and transmission costs when processing all candidate ads per request. This is also a primary reason why advertising systems are constrained to adopt MCA for gradual filtering. Second, repeated RPC communications between feature services and multiple stages incur substantial latency overhead, particularly when handling high-frequency requests in real-time advertising systems. These inefficiencies fundamentally constrain model expressiveness and system scalability.

To address these challenges, we propose the Hybrid Feature Service (HFS), a computation-aware paradigm that achieves more efficient feature processing through categorized feature storage, as shown in Figure 3 (b). The system handles two distinct types of features with different optimization strategies. Candidate ad features, which are numerous in volume, are updated relatively infrequently. To address the network bandwidth challenge posed by $\sim 10^5$ candidate ad features, we implement a local storage solution using memory and SSDs, effectively reducing RPC overhead. Conversely, user-side features update at high frequency but only need to be calculated once per request. To accommodate this pattern efficiently, we store these dynamic features in the Remote Feature Service (RFS) and access them through RPC calls, ensuring real-time data availability while maintaining system performance.

For candidate ads, HFS retains only core categorical feature while omitting numerical and explicit cross-features. This simplification reduces the per-ad feature dimensionality and the whole feature space, enabling storage of ad embeddings through Local Feature Service (LFS). The first-order representation of ad a_i is:

$$\boldsymbol{a}_i = \text{Concat}\Big(\text{Emb}(f_1(a_i)), \text{Emb}(f_2(a_i)), \ldots, \text{Emb}(f_{N_f}(x_i))\Big), \tag{5}$$

where N_f denotes the number of preserved categorical features, $\text{Emb}(\cdot)$ denotes the embedding layer and $\text{Concat}(\cdot)$ is the concatenation operation. We stack $\boldsymbol{x}_i$ together into matrix:

$$\mathbf{E}_{ad} = \big[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N\big], \quad \mathbf{E}_{ad} \in \mathbb{R}^{N \times d}. \tag{6}$$

For user modeling, HFS consolidates all user-specific features (categorical attributes and behavior sequences) and context features (e.g., click time, location) into a single RPC call per request. These unified user and context embeddings are then broadcast to all candidate ads, eliminating redundant feature retrievals while maintaining contextual awareness. Formally, the representation of user is symbolized as $\boldsymbol{x}_i$, and the stacked representation of user behavior sequence is expressed as $\mathbf{E}_{bhvr} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L] \in \mathbb{R}^{L \times d}$, with L representing the length of sequence.

Crucially, HFS shifts computational complexity from feature engineering to neural architecture design – omitted numerical and cross-features are implicitly recovered through subsequent ineteration modules rather than explicit storage.

3.3 RecFormer

Sequence modeling plays a pivotal role in effectively modeling both user interests and item representations. However, conventional approaches face prohibitive computational complexity that scales superlinearly with sequence length and model depth, creating substantial scalability bottlenecks. To address these limitations, we propose RecFormer, a novel framework featuring: (1) a Global Cluster-Former (GCF) that efficiently models intra-sequence relationships, and (2) a Mid-fusion Interest-Former (MIF) for effective cross-sequence mutual information extraction.

Global Cluster-Former Existing approaches for modeling mutual influences among user interaction history or candidate ads typically rely on self-attention mechanisms in Transformer architectures [25, 37]. While effective for small-scale interactions, directly applying Transformer blocks to industrial-scale advertising scenarios with $L = 10^3$ user behavior data and $N = 10^5$ candidate ads per request incurs prohibitive computational complexity of $O(N^2 d)$[1]. This quadratic scaling renders conventional Transformers impractical for real-time inference, as the latency and memory overhead become unsustainable.

To address these challenges, we propose the Global Cluster-Former (GCF) module, which strategically reduces computational complexity while preserving critical externality patterns. GCF consists of m identical layers, each of which can be divided into two

[1] $O(L^2 d)$ for user behavior data.

sub-layers connected by residual connections [6] and layer normalization[2]. Take the candidate ads as an example. The foundational and primary sub-layer is the **cluster-attention**. Concretely, GCF firstly projects input hidden states $\mathbf{H}_{ad}^{\ell}$ (ℓ denotes the ℓ_{th} layer, particularly, $\mathbf{H}_{ad}^{0} = \mathbf{E}_{ad}$) from previous layer into queries, keys and values $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. To reduce computational complexity, GCF employs a clustering mechanism $G(\cdot)$ that aggregates the original N keys and values into N_c cluster-level representations. This approach computes attention from the query $\mathbf{Q}$ to these aggregated keys and values, rather than attending to all N candidates directly. Then, the output representation of this sub-layer is obtained by calculating the multi-head attention (MHA) [25] among the original queries and clustered keys and values.

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = Split\Big(\phi(f_1(\mathbf{H}_{ad}^{\ell}))\Big) \in \mathbb{R}^{N\times d} \tag{7}$$

$$\mathbf{K}' = G(\mathbf{Q}, \mathbf{K}, \mathbf{S}),\ \mathbf{V}' = G(\mathbf{Q}, \mathbf{V}, \mathbf{S}), \quad \mathbf{K}', \mathbf{V}' \in \mathbb{R}^{N_c\times d} \tag{8}$$

$$\text{head}_i = \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}') = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}'^{\top}}{\sqrt{d}}\right)\mathbf{V}' \tag{9}$$

$$\mathbf{H}_{attn}^{\ell} = \phi(f_2(\text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_{N_h}))) \tag{10}$$

where f_1 indicates an MLP that projects X from d to $3d$ hidden nodes, and $Split(\cdot)$ partitions the $3d$-dimensional vector into three distinct components. ϕ denotes the Dice activation [36].

The core innovation of clustering mechanism $G(\cdot)$ lies in the application of learnable cluster matrix $\mathbf{S} \in \mathbb{R}^{N\times N_c}$ [24], which dynamically groups $\mathbf{K}, \mathbf{V}$ into semantically coherent clusters. The cluster matrix serves as a classifier, functioning as proxies to identify semantically similar keys and values within the embedding space. Formally, we begin with calculating surrogate tokens $\mathbf{A_q}, \mathbf{A_k}, \mathbf{A_v}$ for $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ using the cluster matrix. These surrogate tokens are aggregated using adaptive ratios ϕ, which are computed through a linear transformation of the inner product between queries and the corresponding surrogate key or value tokens.

$$\mathbf{A}_q = \mathbf{Q}\mathbf{S}^T, \mathbf{A}_k = \mathbf{K}\mathbf{S}^T, \mathbf{A}_v = \mathbf{V}\mathbf{S}^T \quad \in \mathbb{R}^{N_c\times d} \tag{11}$$

$$\varphi_k = \sigma\big(f_{k_1}(\mathbf{A}_q\mathbf{A}_v^{\top})\big),\ \varphi_v = \sigma\big(f_{v_1}(\mathbf{A}_q\mathbf{A}_k^{\top})\big) \quad \in \mathbb{R}^{N_c\times 1} \tag{12}$$

$$\mathbf{K}' = \varphi_k \odot f_{k_2}(\mathbf{A}_q) + (1-\varphi_k)\odot f_{k_2}(\mathbf{A}_k) \quad \in \mathbb{R}^{N_c\times d} \tag{13}$$

$$\mathbf{V}' = \varphi_v \odot f_{v_2}(\mathbf{A}_q) + (1-\varphi_v)\odot f_{v_2}(\mathbf{A}_v) \quad \in \mathbb{R}^{N_c\times d} \tag{14}$$

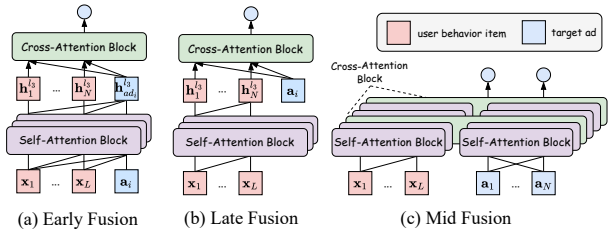
where $f_{k_1}, f_{v_1} : \mathbb{R}^{N_c\times d} \to \mathbb{R}^{N_c\times 1}$ and $f_{k_2}, f_{v_2} : \mathbb{R}^{N_c\times d} \to \mathbb{R}^{N_c\times d}$ are neural networks, $\sigma(\cdot)$ denotes sigmoid function, $\odot$ means eletment-wise multiplication. This design reduces computational complexity from $O(N^2d)$ to $O(NN_cd)$, where $N_c \ll N$. Moreover, GCF exhibits excellent scalability and can easily enhance model performance by stacking multiple blocks. Finally, GCF generates the second-order representations of all candidate ads candidate ads.

The second sub-layer is a feedforward neural network $f_{out} : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$ that processes the output from the cluster-attention sub-layer and generates the final output of the layer.

$$\mathbf{H}_{ad}^{\ell+1} = f_{out}(\mathbf{H}_{attn}^{\ell}) \tag{15}$$

[2]Residual connections and layer normalization are omitted for brevity in the following section.

The final representations of candidate ads are extracted from the last layer, i.e., $\mathbf{H}_{ad} = \mathbf{H}_{ad}^{m}$. User behavior sequence $\mathbf{H}_{usr}$ is computed in a similar way through Eq. (7∼15).



(a) Early Fusion (b) Late Fusion (c) Mid Fusion

Figure 4: Three paradigms for user interest modeling.

Mid-fusion Interest-Former Existing approaches [14, 36] for user interest modeling based on user's behavior sequence predominantly adopt two paradigms: **Early Fusion** and **Late Fusion**. As shown in Figure 4, Early Fusion appends the target ad to the user's behavior sequence and processes the concatenated sequence through m self-attention blocks. While this paradigm enables full interaction between the target ad and user behaviors, it incurs prohibitive computational complexity of $O(mN(L+1)^2d)$ due to repeated sequence processing for each candidate ad. Conversely, Late Fusion first extracts user behavior features independently and later interact them with target ad. Although its complexity reduces to $O((NL+mL^2)d)$, the decoupled interaction limits performance, as user behavior modeling remains agnostic to specific ad during feature extraction. This trade-off between computational efficiency and modeling fidelity poses a critical challenge in large-scale industrial scenarios, where both latency constraints and prediction accuracy are paramount.

To address these limitations, we propose the **Mid-fusion Interest-Former** (MIF) module that strategically balances interaction granularity and computational overhead. Given the hidden representations of user behavior sequence $\mathbf{H}_{usr}$ and candidate ads $\mathbf{H}_{ad}$ generated by GCF, MIF leverages intermediate hidden states from m_c evenly spaced transformer blocks (e.g., blocks $\{m_c, 2m_c, \ldots, m\}$) to perform cross-interactions. MIF also employs cluster attention except that $\mathbf{Q}$ and $\mathbf{K}, \mathbf{V}$ are derived from difference sequences. Conventional approaches employ **target attention** [36], where the query derived from the candidate ad attends to the user behavior sequence to evaluate the relevance of target items within the user's historical interactions. Additionally, we propose **context attention**, which reverses this directional flow by allowing attention signals to propagate from user behaviors to the candidate ad set. This mechanism facilitates capturing deeper user interests within the competitive advertising environment.

Through interval fusion and cluster attention, MIF reduces computational complexity to $O(m_k(N+L)N_cd)$[3]. Assuming that $m = 10, m_c = 2, N = 10^5, L = 10^3, N_c = 100$, MIF requires only $1/10^4$ the FLOPs of early fusion and half the FLOPs of late fusion.

[3]$m_k = \left\lceil \frac{m_2}{m_c} \right\rceil$, where $\lceil\cdot\rceil$ denotes the ceiling function that rounds up to the nearest integer. The proof is shown in Appendix A.1.

After obtaining the representations of candidates ads $\mathbf{H}_{ad} \in \mathbb{R}^{N\times d}$, MIF incorporates various auxiliary tasks (e.g., click, purchase) to accelerate representation learning. Each task is built upon a shared bottom but employs separate parameters for learning. Taking the pCTR prediction as an example:

$$\hat{q}_i^{\text{ctr}} = \text{MLP}_{\text{ctr}}\big(\text{Concat}(\boldsymbol{h}_i, \boldsymbol{e}_u)\big), \quad \forall \boldsymbol{h}_i \in \mathbf{H}_{ad}, i \in [1, N], \tag{16}$$

where $\hat{q}_i^{\text{ctr}}$ represents the externality-aware pCTR of x_i, and MLP$(\cdot)$ refers to a Multi-Layer Perceptron with a sigmoidal activation in the final layer. The training process details will be discussed in Section 4.

3.4 AucFormer

Traditional auction mechanisms like GSP [4] fail to model externalities effectively and lack deep integration with machine learning technologies. In this paper, UniROM introduces the AucFormer framework, comprising 1) a non-autoregressive (NAR) generator that employs a matching mechanism to simultaneously predict allocation probabilities for candidate ads across multiple interface slots [25], and 2) a permutation-aware evaluator that leverages exposure order information to more accurately estimate user engagement metrics, e.g. pCTR, pCVR.

Non-autoregressive generator. The core idea behind AucFormer lies in using slot tokens as surrogate representations. After processing through several GCF layers to derive slot representations $\mathbf{T} \in \mathbb{R}^{K\times d}$, the allocation matrix $\mathbf{A} \in \mathbb{R}^{N\times K}$ is computed as:

$$\mathbf{A} = \mathbf{H}_{ad}\, \mathbf{T}^{\top}$$

where K is the number of slots to be assigned. To align the allocation strategy with the platform's profitability objective, we introduce a bid bias to compute the allocation probability for each candidate ad i at slot k:

$$\text{z}_{i;k} = \text{Softmax}\Big([e^{w_z} \times \hat{q}_j^{\text{ctr}} \times b_j + \mathbf{A}_{j,k}]_{j=1}^{K} \Big)_i, \tag{17}$$

where w_z represents a parameter that can be learned, with the constraint that e^{w_z} remains positive. This constraint ensures that a higher bid results in a higher probability of allocation. The IC proof related to the allocation Equation (17) follows the derivation theory in CGA. During inference, the generator selects the top K ads $Y = [a_{y_1}, a_{y_2}, \ldots, a_{y_K}]$ with the highest scores for each slot. When generating the final winning ad sequence, the generator masks out the ads already assigned to previous slots. For example, when allocating ad for slot k, the selected one is determined by $i = \arg\max_j(z_{j,k}), j \neq [y_1, y_2, \cdots, y_{k-1}]$.

Permutation-aware evaluator. The evaluator aims to predict the permutation-aware values for each ad in the ad allocation. Given an ad allocation $Y = [a_{y_1}, a_{y_2}, \ldots, a_{y_K}]$, the evaluator combines the corresponding high-order representation $\mathbf{H}_{ad}^{Y} = [\boldsymbol{h}_{y_1}, \boldsymbol{h}_{y_2}, \ldots, \boldsymbol{h}_{y_K}]$ with slot embeddings $\mathbf{E}_t$, which is then fed into m_e layer cluster-attention based blocks $\mathcal{T}$ to compute the permutation-aware pCTRs .

$$q_{y_i}^{ctr} = \text{MLP}_{\text{E}}\Big(\mathcal{T}\big(\text{Concat}(\boldsymbol{h}_{y_i}^{a}, \boldsymbol{t}_i, \boldsymbol{e}_u)\big)\Big), \forall i \in [K], \tag{18}$$

where $\mathcal{T}(\cdot)$ is determined by the equations (7) through (14).

Payment network. Motivated by the successful application of neural networks to payment in [37], AucFormer introduces a payment network to learn the optimal payment rule. Specifically, to satisfy IR constraint, the payment network employs a sigmoidal activation function to compute the payment rate $\tilde{\boldsymbol{p}} \in [0, 1]^K$, and subsequently outputs the payment $\boldsymbol{p} = \tilde{\boldsymbol{p}} \odot \boldsymbol{b}$. The payment rate of x_{y_i} in Y is:

$$\tilde{p}_{y_i} = \text{MLP}_{\text{pay}}\Big(\text{Concat}\big(\boldsymbol{h}_{y_i}^{a}, q_{y_i}^{\text{ctr}}, \boldsymbol{b}_{-y_i}\big)\Big), \forall i \in [K], \tag{19}$$

where $\boldsymbol{h}_{y_i}^{a}$ represents the high-order representation of i-th ad in allocation Y, $q_{y_i}^{\text{ctr}}$ denotes the permutation-aware pCTR of i-th ad in Y, and $\boldsymbol{b}_{-y_i} \in \mathbb{R}^{K-1}$ is the bids vector of ads in Y excluding the i-th ad. It is evident that the payment network operates independently of the evaluator. During inference, only the generator and the payment network are deployed to reduce computation costs.

4 TRAINING AND OPTIMIZATION

In this section, we present a two-stage training framework inspired by large language model optimization [20]. The first stage, pre-training, focuses on aligning UniROM with user engagement signals such as clicks and purchases. Subsequently, the post-training stage employs reinforcement learning to optimize UniROM's response to auction feedback and guide the payment network under predefined economic constraints.

4.1 Pre-training

We treat each request as a sample κ to build the pre-training dataset $\mathcal{D}$, rather than considering each exposure as a sample. Each sample κ contains N_s unexposed ads selected via popularity sampling [19] from the valid candidate ad pool, combined with K exposed ads from actual impressions. The popularity sampling strategy prioritizes frequently occurring candidates while maintaining long-tail coverage through probabilistic selection, thereby enhancing training efficiency without sacrificing representation diversity.

In each sample κ, we utilize the user's clicks under current request as labels for the K exposed ads and the user's overall click behaviors across the platform as labels for N_s unexposed ads, denoted as $\zeta_i^{clk} \in \{0, 1\}$. The overall click behaviors on one specific ad include the user's clicks from other entries on the current platform as well as the user's clicks when the ad is displayed in its organic item form in one same session. The pre-training objective minimizes binary cross-entropy loss over both the set-aware pCTR:

$$\mathcal{L}_{pt} = -\frac{1}{|\mathcal{D}|} \sum_{\kappa\in\mathcal{D}} \sum_{i=1}^{K+N_s} \Big(\zeta_i^{clk} \log(\hat{q}_i^{\text{ctr}}) + (1 - \zeta_i^{clk}) \log(1 - \hat{q}_i^{\text{ctr}}) \Big). \tag{20}$$

For brevity, we only utilize click signals for illustration. Additionally, other user behavioral signals, like purchases, reviews, and so forth, can also contribute to the representation learning through pre-training in a similar manner.

4.2 Post-training

During the post-training phase, reinforcement learning is primarily used to fine-tune the model in alignment with the platform's profitability objectives. This process involves three key components: training the reward model, reinforcement learning from auction feedback, and optimizing the payment network.

Training the Reward Model The permutation-aware evaluator in AucFormer serves as a reward model, designed to evaluate the quality of the generated ad sequence. To improve training effectiveness and prevent unnecessary repetition, we utilize the pre-trained model's parameters and freeze the parameters of the modules before RecFormer during the reward model training. Formally, we only use the K exposed ads from each sample κ to train the reward model and the loss is calculated as:

$$\mathcal{L}_{rm} = -\frac{1}{|\mathcal{D}|}\sum_{\kappa\in\mathcal{D}}\sum_{i=1}^{K}\Big(\zeta_i^{\kappa}\log(q_i^{ctr}) + (1-\zeta_i^{\kappa})\log(1-q_i^{ctr})\Big). \tag{21}$$

Reinforcement Learning from Auction Feedback. Following reward model convergence, the non-autoregressive generator undergoes optimization via Reinforcement Learning from Auction Feedback (RLAF) with frozen evaluator parameters. The reward signal r_{y_i} for ad a_{y_i} in sequence $Y = [a_{y_1}, a_{y_2}, ..., a_{y_K}]$ quantifies its marginal contribution to platform revenue:

$$r_{y_i} = \sum_{a_{y_i}\in Y} b_{y_i} q_{y_i}^{ctr} - \sum_{a_{y_j}\in Y_{-i}} \tilde{b}_{y_j}\tilde{q}_{y_j}^{ctr}, \quad \forall i \in [K], \tag{22}$$

where Y_{-i} represents the best ad sequence excluding a_{y_i}, and $\tilde{b}_{y_j}\tilde{q}_{y_j}^{ctr}$ indicates the corresponding bid and pctr. The policy gradient objective maximizes expected rewards through:

$$\mathcal{L}_{rlaf} = -\frac{1}{|\mathcal{D}|}\sum_{\kappa\in\mathcal{D}}\sum_{Y\in\mathcal{X}(\kappa)}\sum_{i=1}^{K}\Big(r_{y_i}\log(z_{y_i;i})\Big), \tag{23}$$

where $\mathcal{X}(\kappa)$ denotes the valid candidate ads in sample κ and Y denotes the ad sequence output by the non-autoregressive generator. As the needs of advertisers and business goals constantly evolve, we decided to freeze the parameters of RecFormer during RLAF, focusing solely on optimizing the parameters of AucFormer that are closely tied to the business objectives.

Optimization of Payment Network. The payment network optimization follows generator-evaluator training, employing a Lagrangian dual formulation to balance revenue maximization and IC constraints [37]. The loss function incorporates both platform revenue and ex-post regret minimization:

$$\mathcal{L}_{pay} = -\frac{1}{|\mathcal{D}|}\sum_{\kappa\in D}\sum_{Y\in\mathcal{X}(\kappa)}\left(\sum_{x_{y_i}\in Y} p_{y_i}^{Y} q_{y_i}^{ctr} - \sum_{x_{y_i}\in Y}\lambda_{y_i}\widehat{\mathrm{tgt}}_{y_i} - \frac{\rho}{2}\sum_{x_{y_i}\in Y}(\widehat{\mathrm{tgt}}_{y_i})^2\right), \tag{24}$$

where $\widehat{\mathrm{tgt}}_{y_i}$ is the *ex-post regret* for ad x_{y_i} in Y, λ_{y_i} is the Lagrange multiplier used to balance revenue and IC constraints and ρ is the hyperparameter for the IC penalty term.

5 EXPERIMENTS

In this section, we evaluate the effectiveness of UniROM using offline experiments and online A/B testing.

5.1 Experiment Setup

5.1.1 Dataset. We provide empirical evidence for the effectiveness of UniROM on the industrial dataset **Meituan** [4] from a well-known local life platform that leverages Location-Based Services (LBS) to provide users with convenient and tailored services. The industrial dataset is constructed from the real logs of advertising platform from April 2024 to October 2024. It contains 200 million requests from more than 2 million users across nearly 10 million ads. The first 200 days are used for pre-training, a randomly sampled 50 days subset is employed fro post-training, and the last 14 days are used as the test set. In the LBS scenario, the maximum number of valid candidate ads in each sample is $N = 100,000$. To enhance training efficiency, popularity sampling is utilized during training, with $N_s = 2995$ and $K = 5$. During testing, inference is conducted using all candidate ads for each sample, where $N = 100,000$ and $K = 5$ are employed.

5.1.2 Evaluation Metrics. In offline experiments, we utilize standard ranking metrics including Recall@50 and Area Under the Curve (AUC) to evaluate the pre-ranking and ranking stages, respectively. We further employ an offline replay system with the permutation-aware evaluator to evaluate the expected CTR and the expected revenue of the results generated by different architectures.

- eCTR = $\frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{\kappa\in\mathcal{D}_{\text{test}}}\sum_{Y\in\mathcal{X}(\kappa)}\sum_{x_{y_i}\in Y} q_{y_i}^{ctr}\times 100\%$.
- eRPM = $\frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{\kappa\in\mathcal{D}_{\text{test}}}\sum_{Y\in\mathcal{X}(\kappa)}\sum_{x_{y_i}\in Y}\left(q_{y_i}^{ctr} * p_{y_i}\right)\times 1000$.
- IC Metric: $\Psi = \frac{1}{|\mathcal{D}_{\text{test}}|}\sum_{\kappa\in\mathcal{D}_{\text{test}}}\sum_{i\in k}\frac{\widehat{\mathrm{tgt}}_i^{\kappa}}{u_i(v_i^{\kappa};\boldsymbol{b}^{\kappa})}$, where $\widehat{\mathrm{tgt}}_i^{d}$ denotes the empirical ex-post regret for advertiser i in session data κ, and u_i is the realized utility. This metric evaluates IC, representing the relative utility gain an advertiser could obtain by manipulating its bid [17]. Following [37], IC is empirically tested via counterfactual perturbation: for each advertiser, the bid b_i is replaced with $\gamma \times b_i$, where $\gamma \in \{0.2 \times j \mid j = 1, 2, \ldots, 10\}$.

In online A/B testing, we introduce the following three metrics to measure platform revenue, user experience, and Return on Investment of advertisers, respectively.

- **Revenue Per Mille**: RPM = $\frac{\sum \text{click}\times\text{payment}}{\sum \text{impression}}\times 1000$.
- **Click-Through Rate**: CTR = $\frac{\sum \text{click}}{\sum \text{impression}}\times 100\%$.
- **Return on Investment**: ROI = $\frac{\sum \text{gross merchandise volume}}{\sum \text{payment}}$.

5.1.3 Baselines. We compare UniROM with the following two representative architectures, which are widely used in industry: 1) **MCA**. The Multi-stage Cascading Architecture is a common design for online advertising systems. To implement this architecture effectively, we employ four representation methods: SASREC [10] for recall, DSSM [8] for pre-ranking, DIN [36] for ranking, and GSP [4] for auction. 2) **FS-LTR**. Full Stage Learning to Rank [35] is a unified training framework designed for multi-stage recommendation systems. It leverages relabeled data from all stages to train models in MCA, ensuring that the top-ranked items are more likely to pass through subsequent stages and align with user interests. To ensure effectiveness and comparability of FS-LTR, we use the same architecture as MCA, incorporating FS-LTR's sample re-labeling technique during training to enhance consistency.

[4]Due to the need to protect business secrets, some transformations were applied to the results. These transformations were carefully designed to maintain the statistical properties while ensuring that no sensitive business - related information could be reverse - engineered from the transformed results.

5.1.4 Hyperparameters. For MCA and FS-LTR, we follows the hyperparameters setting in [16]. For UniROM, we tried different hyperparameters using grid search. Due to space limitations, only the most optimal parameters are presented in this paper. The clustering approach G employs an adaptive clustering mechanism, the hidden layers of the MLP are 128 and 32, the learning rate is 10^{-3}, the batch size is 128 and the optimizer is Adam. the user behavior sequence length $L = 1000$, the dimension size $d = 128$, the layer number of GCF and MIF $m = 6$, the layer number of RecFormer $m_e = 3$, the number of interval layers in MIF $m_k = 2$, the number of attention heads $N_h = 4$ and the number of adaptive clusters $N_c = 128$.

5.2 Offline Experiments

Table 1 summarizes the results on industrial dataset from the offline experiments. Each offline experiment is repeated 5 times with different random seeds and each result is presented in the form of mean ± standard. The experimental results yield the following observations. On the industrial dataset, UniROM improves over the state-of-the-art baselines in AUC, Recall@50, eCTR, eRPM, and Ψ respectively, demonstrating the superiority of UniROM. Specifically, UniROM achieves a Recall@50 of 0.513, representing a significant lift of 20.4%[5] over the strongest baseline FS-LTR. The AUC increases to 0.754 (+1.48%), and eCTR improves +8.3%, demonstrating superior user modeling and ranking capabilities. In terms of platform revenue, UniROM achieves an eRPM of 217.1, which is 11.4% higher than FS-LTR. Most notably, the IC metric Ψ is dramatically reduced to 2.3%, compared to 9.1% and 9.4% for FS-LTR and MCA respectively, indicating much stronger incentive compatibility and robustness to auction constraints.

Table 1: The experimental results of different architectures.

Method	Recall@50	AUC	eCTR(%)	eRPM	Ψ
MCA	0.289	0.741	6.043	185.2	9.4%
FS-LTR	0.426	0.743	6.140	194.9	9.1%
UniROM	**0.513**	**0.754**	**6.652**	**217.1**	**2.3%**

5.3 Ablation Study

To verify the effectiveness of UniROM's various design considerations, we construct three variants:

- **UniROM**$_{-gcf}$ removes the GCF and uses point-wise pCTR instead of set-aware pCTR for next tasks.
- **UniROM**$_{-mif}$ removes the MIF and only uses DIN to capture user's interests from user's historical behavior sequence.
- **UniROM**$_{-auf}$ removes the AucFormer and uses GSP mechanism for completing ad allocation and payment.

Judging from the online experimental results in Table 2, we have the following findings: 1) The variant without GCF performs worse than UniROM. This phenomenon proves that our proposed GCF can effectively model the global set-aware externalities and help UniROM to achieve better performance. Especially in our LBS scenarios, the benefits of this global externalities modeling are further enhanced. 2) The experimental results of **UniROM**$_{-mif}$ are worse than UniROM, this supports that the deep modeling of user behavior sequences is crucial for both prediction and generation tasks. (3) The performance gap between w/ and w/o AucFormer is obvious. This indicates that AucFormer structure and RLAF training approach can effectively align with auction preferences, resulting in improved ad sequence generation and overall performance.

Table 2: Performance comparison of different variants.

Method	eCTR(%)	eRPM
UniROM	**6.652**	**217.1**
UniROM-gcf	6.466 (-1.3%)	214.8 (-1.1%)
UniROM-mif	6.389 (-2.5%)	212.5 (-2.1%)
UniROM-auf	6.415 (-2.1%)	208.0 (-4.2%)

5.4 Cross Features Analysis

This section demonstrates how target-attention (TA) and context-attention (CA) mechanisms in MIF module mitigate performance loss from cross-feature elimination. Furthermore, we also compare the results with the Late Fusion (LF), where the MIF module is replaced by a late fusion approach. In our experiment, we remove 18 cross features (CF) out of the initial 94 features, and the corresponding results are shown in Table 3.

Table 3: Impact analysis of discarding cross-features on AUC and Deviation. Values in the parentheses of the first block show the difference from the underlined value, and the second block shows the difference from the bold value. ↓ indicates that smaller is better.

Method	AUC	Dev.↓
MCAs	<u>0.741</u>	<u>0.112</u>
MCAs w/o CF	0.733 (-0.8%)	0.131 (+1.9%)
UniROM	**0.754**	**0.005**
UniROM w/o CF	0.751 (-0.3%)	0.011 (+0.6%)
UniROM$_{\text{-CA}}$ w/o CF	0.745 (-0.9%)	0.036 (+3.1%)
UniROM$_{\text{-TA}}$ w/o CF	0.743(-1.1%)	0.047 (+4.2%)
UniROM$_{\text{-mif}}$ w/o CF	0.737 (-1.7%)	0.048 (+4.4%)
UniROM$_{\text{LF}}$ w/o CF	0.741 (-1.3%)	0.037 (+3.2%)

As illustrated in the table, cross-features demonstrate a significant impact on both order prediction (AUC) and accuracy (Dev.)[6] performance. Notably, their influence on MCAs is substantially greater than on UniROM. When cross-features are omitted, the removal of both context-attention and target-attention mechanisms leads to considerable AUC degradation, with reductions of 0.9% and 1.1%, respectively. More critically, removing both components (MIF) results in performance deterioration that nearly equals the combined effect of their individual removals, indicating that these

[5]Lift percentage means the improvement of UniROM over the best baselines

[6]Dev. means deviation, which measures the discrepancy between predicted click-through rate (pCTR) and actual click-through rate (CTR). Formally, Dev. $= |1 - \frac{\text{pCTR}}{\text{CTR}}|$

features contribute synergistically rather than independently to model performance. Moreover, replacing MIF with late fusion degrades the AUC by 1.0% and increases the deviation by 2.6%. The empirical results demonstrate that MIF robustly addresses the performance drop attributable to cross-feature information loss.

5.5 Scaling Law

In UniROM, each module uses stacked blocks based on cluster-attention as core structure to ensure scalability. Here, we vary the number of blocks across modules to study scaling laws. Besides, we Results in Figure 5 reveal:

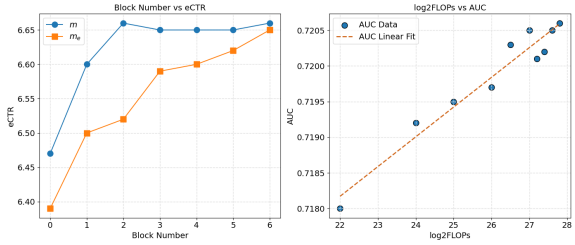


Figure 5: Scaling trends featuring the relationship between model layer depth and FLOPs.

1) The performance curve of and m and m_e illustrates that increasing the number of blocks in RecFormer and AucFormer yields progressive performance gains, though with diminishing returns beyond a critical threshold. This plateau effect likely stems from inherent constraints imposed by fixed data scales and sequence lengths in our experimental setup. While exploring larger datasets and extended sequences could unlock additional scaling potential, such investigations remain beyond the scope of this study due to resource constraints.

2) The figure on the right reveals a clear positive correlation between computational cost and model performance in UniROM. The upward-trending linear fit (brown dashed line) indicates that increased computational resources consistently enhance recommendation accuracy, though the marginal gains gradually diminish at higher FLOPs levels. This suggests that while scaling computation remains an effective strategy for performance improvement within the tested range, the model's architecture or data constraints may eventually limit further gains—mirroring the saturation pattern observed in block stacking. The relationship underscores the importance of balancing computational investment against diminishing returns in practical deployment scenarios.

5.6 Online Results

To verify UniROM's effectiveness in the real-world, we compare UniROM with the fully deployed MCA in industrial advertising system through online A/B tests. Table 4 presents the results of online A/B testing conducted from November 18 to November 24, 2024. Experimental results demonstrate 3.8% improvements in CTR, 11.2% in RPM, and 3.1% in ROI respectively, indicating UniROM's effectiveness in enhancing user experience, platform revenue, and advertiser utility. Notably, the harmonized gains in user engagement and advertiser's ROI suggest that revenue growth stems not from payment inflation, but from enhanced modeling capabilities.

We also analyzed the online inference computational complexity of UniROM and MCA. UniROM processes hundreds of times more candidate ads than MCA's ranking model with only a 5 ms average increase (2.2% relatively) in online response time (RT) per request). The complexity is discussed in Appendix A.1.

Table 4: Experimental results from Online A/B tests.

Relative change in metrics	CTR	RPM	ROI	RT
UniROM over baseline-MCA	+5.2%	+13.6%	+3.1%	+2.2%

Recent efforts like FS-LTR [35] and COPR [33] attempt to enhance stage consistency through unified training with relabeled multi-stage data. While demonstrating improved performance, such methods remain constrained by the MCA paradigm's structural limitations: 1) The inherent conflict between stage-specific optimization objectives creates irreducible prediction inconsistencies; 2) The sequential filtering mechanism prevents holistic externality modeling across all candidates; 3) Feature engineering dominance limits scalability compared to computation-driven architectures [30]. Notably, Zheng et al. [35] developed an enhanced ranking principle to mitigate selection bias in downstream stages, representing the state-of-the-art MCA improvement. However, their solution still requires complex multi-stage coordination rather than offering true end-to-end optimization. These fundamental limitations motivate our investigation into alternative architectural paradigms.

6 CONCLUSIONS

In this paper, we propose UniROM that **Uni**fies the entire ad ranking and allocation pipeline as **O**ne **M**odel. Our key innovations include: 1) a hybrid feature service that decouples user and ad feature processing to reduce latency while preserving expressiveness; 2) an transformative recommendation framework that leverages an innovative cluster-attention mechanism to efficiently model users' deep interests and contextual externalities; and 3) a bi-stage training strategy align with user preference and platform objectives. Extensive experiments on public and industrial datasets demonstrate UniROM's superiority over state-of-the-art approaches. In the future, we will further explore scaling laws and different training paradigms to enhance UniROM's capabilities.

A APPENDIX

A.1 Complexity Analysis

We provide a detailed comparison of computational complexity between MCA and UniROM. All floating-point operations (FLOPs) are calculated based on standard Transformer block operations [25]. Given sequence length L, hidden dimension d, the total FLOPs of a standard Transformer block consist of three components:

$$\begin{aligned}\text{FLOPs}_{block} = &\underbrace{4 \times 2Ld^2}_{\text{matrix calculation}} + \underbrace{16Ld^2}_{\text{feed-forward network}} + \underbrace{4L^2d}_{\text{attention}} \\ &= 4L^2d + 24Ld^2.\end{aligned} \tag{25}$$

Similarly, when we change the query to $\mathbf{Q} \in \mathbb{R}^{L_1 \times d}$ and key,value to $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{L_2 \times d}$, the total FLOPs of the Transformer block is:

$$\text{FLOPs}_{block_2} = 4L_1L_2d + 20L_1d^2 + 4L_2d^2. \tag{26}$$

MCA Complexity. The FLOPs of MCA mainly has two parts:

(1) **Pre-ranking Stage**: Using the Dual-tower model [8] as a reference, the primary operations involve the computation of the user embedding ($L \times d \times d \times 2$) and the execution of the inner dot product for N candidate ads ($N \times d$):

$$\text{FLOPs}_{pre} = Nd + 2Ld^2. \tag{27}$$

(2) **Ranking Stage**: Taking a Transformer-based model for instance, we use Early Fusion paradigm to compensate for the efficiency decline in MCA. The Flops is:

$$\text{FLOPs}_{rank} = m_r\left[4N_r(L+1)^2d + 24N_r(L+1)d^2\right]. \tag{28}$$

where m_r is the number of blocks, N_r is the number of valid candidate ads in ranking stage, where $N_r < N$.

Finally, the total Flops of MCA is calcalated as:

$$\text{FLOPs}_{mca} = \text{FLOPs}_{pre} + \text{FLOPs}_{rank}. \tag{29}$$

UniROM Complexity. The FLOPs of UniROM contains three modules:

(1) **Global-Cluster Former Module** (m blocks):

$$\text{FLOPs}_{gcf} = m(6NN_cd + 4NN_cd + 24Nd^2). \tag{30}$$

where N_c is the number of surrogate tokens and $N_c \ll N$, $6NN_cd$ is the Flops of cluster mechanism G and $24Nd^2$ is the Flops of origin matrix calculation.

(2) **Mid-Interest Interest-Former Module** (m_k blocks):

$$\begin{aligned}FLOPs = &\underbrace{m_kNN_cd}_{\text{target attention}} + \underbrace{m_kLN_cd}_{\text{context attention}} \\ &= m_k(N+L)N_cd\end{aligned} \tag{31}$$

(3) **AucFormer Module** (m_e blocks):

$$\begin{aligned}\text{FLOPs}_{auf} = &m_e(4N_a^2d + 24N_ad^2) + \\ &m_e(4KN_ad + 18Kd^2 + 4N_ad^2).\end{aligned} \tag{32}$$

where K and N_a are the number of slots and valid ads.

Finally, the total Flops of UniROM is calcalated as:

$$\text{FLOPs}_{UniROM} = \text{FLOPs}_{gcf} + \text{FLOPs}_{mif} + \text{FLOPs}_{auf}. \tag{33}$$

Both the computational complexities of MCA and UniROM are proportional to the number of candidates(N). With $N_r = \alpha N$, $m_r = m$, and $m = 2m_c$, when N is large, the ratio of proportions can be approximated as follows:

$$\frac{\text{FLOPs}_{UniROM}}{\text{FLOPs}_{mca}} \approx \frac{2m_kNLd^2}{m_rN_r4L^2d}$$

In our case, with $\alpha = 0.033$, this results in a ratio of 0.97. The slight difference in proportions leads to a significant FLOPs gap as the number of candidates increases, demonstrating UniROM's capability and efficiency within our design.

A.2 Auction Constraints

To ensure incentive compatibility (IC) in our model, we adopt the concept of *ex-post regret* [3, 37] to quantify the potential gain an advertiser could obtain by untruthfully reporting their bid. This formulation enables us to enforce IC constraints in a differentiable manner, suitable for end-to-end optimization.

Formally, given the generated sequence Y, ad $x_i \in Y$ with true valuation v_i, the ex-post regret is defined as:

$$\text{tgt}_i(v_i, Y) = \max_{b_i'}\left\{u_i(v_i; b_i', \boldsymbol{b}_{-i}, Y) - u_i(v_i; b_i, \boldsymbol{b}_{-i}, Y)\right\}, \tag{34}$$

where b_i is the truthful bid, b_i' is a potential misreport, and $\boldsymbol{b}_{-i}$ represents bids excluding the item x_i. The IC constraint is satisfied if and only if $\text{rgt}_i = 0$ for all advertisers. In practice, we approximate this using M sampled valuations from distribution $\mathbb{F}$, the empirical ex-post regret for ad x_i is

$$\widehat{\text{tgt}}_i = \frac{1}{M}\sum_{j=1}^{M}\text{tgt}_i(v_i^j, Y). \tag{35}$$

We then formulate the auction design problem as minimizing the expected negative revenue under the constraint that the empirical ex-post regret remains zero for each ad x_i:

$$\min_{\boldsymbol{w}} \ -\mathbb{E}_{\boldsymbol{v}\sim\mathbb{F}}\left[\text{Rev}^{\mathcal{M}}\right], \quad \text{s.t.} \quad \widehat{\text{tgt}}_i = 0, \ \forall i \in [N], \tag{36}$$

With the optimization loss of payment network defined in Equation (24), the proposed UniROM ensures IC approximately.

References

[1] 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.

[2] 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).

[3] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. 2019. Optimal auctions through deep learning. In *International Conference on Machine Learning*. PMLR, 1706–1715.

[4] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* 97, 1 (2007), 242–259.

[5] Siyu Gu and Xiangrong Sheng. 2022. On Ranking Consistency of Pre-ranking Stage. *arXiv preprint arXiv:2205.01289* (2022).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[8] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[9] Bernard J Jansen and Tracy Mullen. 2008. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business* 6, 2 (2008), 114–131.

[10] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[11] Ningyuan Li, Yunxuan Ma, Yang Zhao, Zhijian Duan, Yurong Chen, Zhilin Zhang, Jian Xu, Bo Zheng, and Xiaotie Deng. 2023. Learning-based ad auction design with externalities: the framework and a matching-based approach. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1291–1302.

[12] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 2022. Inttower: the next generation of two-tower model for pre-ranking system. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3292–3301.

[13] Xuejian Li, Ze Wang, Bingqi Zhu, Fei He, Yongkang Wang, and Xingxing Wang. 2024. Deep automated mechanism design for integrating ad auction and allocation in feed. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1211–1220.

[14] Guogang Liao, Xiaowen Shi, Ze Wang, Xiaoxu Wu, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. Deep page-level interest network in reinforcement learning for ads allocation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2292–2296.

[15] Guogang Liao, Ze Wang, Xiaoxu Wu, Xiaowen Shi, Chuheng Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. Cross dqn: Cross deep q network for ads allocation in feed. In *Proceedings of the ACM Web Conference 2022*. 401–409.

[16] Qi Liu, Kai Zheng, Rui Huang, Wuchao Li, Kuo Cai, Yuan Chai, Yanan Niu, Yiqun Hui, Bing Han, Na Mou, et al. 2024. RecFlow: An Industrial Full Flow Recommendation Dataset. *arXiv preprint arXiv:2410.20868* (2024).

[17] Xiangyu Liu, Chuan Yu, Zhilin Zhang, Zhenzhe Zheng, Yu Rong, Hongtao Lv, Da Huo, Yiqing Wang, Dagui Chen, Jian Xu, et al. 2021. Neural auction: End-to-end learning of auction mechanisms for e-commerce advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3354–3364.

[18] Xinchen 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).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[20] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[21] 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.

[22] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. 2022. Rankflow: Joint optimization of multi-stage cascade ranking systems as flows. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 814–824.

[23] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).

[24] Adjorn Van Engelenhoven, Nicola Strisciuglio, and Estefanía Talavera. 2024. CAST: Clustering Self-Attention using Surrogate Tokens for Efficient Transformers. *arXiv preprint arXiv:2402.04239* (2024).

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[26] Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 105–114.

[27] Yunli Wang, Zhiqiang Wang, Jian Yang, Shiyang Wen, Dongying Kong, Han Li, and Kun Gai. 2024. Adaptive Neural Ranking Framework: Toward Maximized Business Goal for Cascade Ranking Systems. In *Proceedings of the ACM on Web Conference 2024*. 3798–3809.

[28] Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. Cold: Towards the next generation of pre-ranking system. *arXiv preprint arXiv:2007.16122* (2020).

[29] Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, et al. 2025. Sparse Meets Dense: Unified Generative Recommendations with Cascaded Sparse-Dense Representations. *arXiv preprint arXiv:2503.02453* (2025).

[30] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, et al. 2024. Actions speak louder than words: trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*. 58484–58509.

[31] Zhixuan Zhang, Yuheng Huang, Dan Ou, Sen Li, Longbin Li, Qingwen Liu, and Xiaoyi Zeng. 2023. Rethinking the role of pre-ranking in large-scale e-commerce searching system. *arXiv preprint arXiv:2305.13647* (2023).

[32] Binglei Zhao, Houying Qi, Guang Xu, Mian Ma, Xiwei Zhao, Feng Mei, Sulong Xu, and Jinghe Hu. 2025. A Hybrid Cross-Stage Coordination Pre-ranking Model for Online Recommendation Systems. *arXiv preprint arXiv:2502.10284* (2025).

[33] Zhishan Zhao, Jingyue Gao, Yu Zhang, Shuguang Han, Siyuan Lou, Xiang-Rong Sheng, Zhe Wang, Han Zhu, Yuning Jiang, Jian Xu, et al. 2023. COPR: Consistency-Oriented Pre-Ranking for Online Advertising. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4974–4980.

[34] 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.

[35] Kai Zheng, Haijun Zhao, Rui Huang, Beichuan Zhang, Na Mou, Yanan Niu, Yang Song, Hongning Wang, and Kun Gai. 2024. Full stage learning to rank: A unified framework for multi-stage systems. In *Proceedings of the ACM Web Conference 2024*. 3621–3631.

[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] Ruitao Zhu, Yangsu Liu, Dagui Chen, Zhenjia Ma, Chufeng Shi, Zhenzhe Zheng, Jie Zhang, Jian Xu, Bo Zheng, and Fan Wu. 2024. Contextual Generative Auction with Permutation-level Externalities for Online Advertising. *arXiv preprint arXiv:2412.11544* (2024).