UniRank: Benchmarking Ranking Models for Unified Sequential Modeling and Feature Interaction

Honghao Li
salmon1802li@gmail.com
Anhui University
Hefei, Anhui Province, China

Xianquan Wang
wxq2001@gmail.com
Independent Researcher
Hefei, Anhui Province, China

Zibin Zhang
bingoozhang@tencent.com
Tencent Inc.
Guangzhou, Guangdong Province
China

Yi Zhang
zhangyi.ahu@gmail.com
Anhui University
Hefei, Anhui Province, China

Kangyi Lin
konyel@163.com
Tencent Inc.
Guangzhou, Guangdong Province
China

Yiwen Zhang*
zhangyiwen@ahu.edu.cn
Anhui University
Hefei, Anhui Province, China

arXiv:2607.19987v3 [cs.IR] 18 Nov 2026

Abstract

Ranking is a core stage in online advertising and recommender systems. Modern ranking models increasingly unify sequential modeling and feature interaction, yet many advances rely on proprietary data, closed implementations, and large-scale industrial infrastructure. This setting limits reproducible comparison and hinders academic study of scaling laws, long-sequence modeling, and multi-task ranking. To address these limitations, this paper proposes **UniRank**, an open benchmark for ranking models that unify sequential modeling and feature interaction. UniRank uses chronological pointwise autoregressive supervision, standardizes evaluation across feedback tasks, and provides a PyTorch toolkit with Distributed Data Parallel training, operator optimization, mixed-precision training, attention optimization, and other efficiency techniques that reduce hardware requirements. We benchmark 15 representative unified ranking models on five public datasets from short-video, advertising, and e-commerce platforms, with the largest containing over 700 million instances and the longest behavior sequence exceeding 10^5 interactions. UniRank provides a reproducible foundation for comparing unified ranking models, studying scaling laws under limited compute, and narrowing the gap between academic and industrial ranking research. We believe UniRank benefits researchers, practitioners, and beginners through reproducible experiments, production-oriented evaluation, and accessible implementations. Code and data are available at https://github.com/salmon1802/UniRank.

Keywords

Ranking Models, Recommender Systems, Multi-Task Learning, Benchmarking, Sequential Modeling, Feature Interaction

1 Introduction

Recommender systems drive user engagement and business value in e-commerce [13, 89], short-video streaming [14], and online advertising [6, 71]. Under large item corpora and strict latency constraints, industrial recommender systems typically adopt a multi-stage pipeline, including *retrieval* [8, 29], *pre-ranking* [15, 73], *ranking* [17, 37], and *re-ranking* [1, 55].

Within this pipeline, the ranking model is the central decision component [17]. Given heterogeneous multi-field features such as user profiles, item attributes, device types, request times, and historical behaviors, the ranking model estimates the probabilities of task-specific feedback events, including clicks, conversions, and other user responses [25, 37, 93]. Consequently, ranking models directly affect user experience and platform revenue, and even small improvements in offline metrics such as the area under the ROC curve (AUC) can correspond to meaningful business gains [6]. The design and optimization of ranking models therefore remain important in both academia and industry [13, 89].

Recent advances push ranking models toward more expressive feature representations and user intent signals. This progression yields three broad methodological families: (1) *The Shallow Learning Method* uses Logistic Regression [61] and factorization-based models [27, 60, 66]. These models are efficient but have limited capacity and rely on manual feature engineering. (2) *The Deep Feature Interaction Method* embeds sparse categorical features and replaces manual crosses with neural interaction modules. Representative examples use MLPs, compressed interaction networks, cross networks, or quadratic neurons to learn non-linear high-order feature combinations, including DeepFM [17], xDeepFM [37], DCNv2 [71], and QNN [31]. (3) *The Sequence Modeling Method* captures temporal changes in user interest. Models such as DIN [89] and DIEN [13] use target-aware attention or recurrent networks to aggregate histories for the target item. To improve performance, deployed ranking models [49, 83] may integrate multiple DCNv2, MLP, and DIN modules. However, these non-unified designs introduce fine-grained operators and execution boundaries that limit parallel scaling [92].

Inspired by large language models [88, 92], current research increasingly adopts token-based unified architectures that combine sequential modeling with feature interaction. These architectures improve Model FLOPs Utilization (MFU) and scale more effectively with sequence length and model capacity. They follow either *stacked* designs [16, 18, 26, 70, 79, 84, 92], which perform sequential modeling before feature interaction, or *layer-wise* designs [3, 11, 23, 24, 32, 41, 85, 90], which integrate both operations within each layer. Together, these architectures provide a unified

*Corresponding author

foundation for studying how sequential modeling and feature interaction scale under a consistent protocol.

Despite this progress, scaling ranking research to industrial conditions remains challenging. Many strong ranking models, large-scale datasets, and training systems rely on proprietary resources, leaving academic researchers with limited access to the data and infrastructure needed to study scaling laws, long-sequence behavior, and task trade-offs under realistic conditions. These barriers constrain academic exploration and systematic comparison.

Existing studies and toolkits provide integrated platforms for recommendation evaluation, including FuxiCTR [25, 93], Scenario-Wise Rec [34], DeepCTR [63], RecBole [87], and MMLRec [80]. These frameworks greatly facilitate standardized research across complementary settings: FuxiCTR and DeepCTR focus on single-task CTR prediction, RecBole covers general and sequential recommendation, and Scenario-Wise Rec and MMLRec emphasize multi-scenario or multi-task learning. However, jointly extending this standardization to industrial-scale long-sequence modeling and unified architectures remains challenging. In particular, no common protocol yet exists for chronological data partitions, behavior sequence construction, capacity-controlled comparison, and evaluation across feedback tasks. Further evidence is needed to understand how sequential modeling and feature interaction scale across feedback signals.

To address these issues, we propose the Unified Ranking Model Benchmark (**UniRank**) for ranking models with unified sequential modeling and feature interaction. Specifically, UniRank evaluates 15 representative models on five large-scale public datasets from short-video, advertising, and e-commerce platforms. It also provides handbook-style controlled experiments on tokenization, activation functions, architectural enhancements, optimizers, sequence length, and model scale. These experiments standardize architectural studies and reduce the cost of rebuilding datasets, implementations, and ablation studies.

This work serves several groups of readers:

- **Researchers**: UniRank provides accessible open-source code and processed datasets for reproducible model development, controlled experimentation, and direct architectural comparison across standardized settings.
- **Practitioners**: UniRank provides acceleration components, supports multi-feedback tasks, and offers production-oriented recipes for efficient training and practical validation.
- **Beginners**: The repository serves as an educational resource with accessible PyTorch implementations and configurations that help readers understand modern unified ranking architectures through hands-on examples.

In summary, our work makes the following main contributions:

- We propose **UniRank**, the first open benchmark dedicated to ranking architectures that unify sequential modeling and feature interaction, enabling systematic comparison under standardized data, optimization, and efficiency settings.
- We integrate Distributed Data Parallel (DDP), operator, mixed-precision, and attention optimizations to reduce hardware costs.
- We release source code, training configurations, processed datasets, and complete results for 15 models on five public datasets, spanning up to 700 million instances and behavior sequences exceeding 10^5 events, supporting reproducible evaluation and future benchmark extensions.

2 Ranking Models

2.1 Input Features and Modeling Paradigms

Ranking models estimate target-item feedback probabilities. UniRank represents each impression with non-sequential features $\mathcal{F}$, behavior sequences $\mathcal{S}$, and target features $\mathcal{V}$:

$$\mathcal{F} = \{f_1, \dots, f_m\}, \quad \mathcal{S} = (s_1, \dots, s_L), \quad \mathcal{V} = \{v_1, \dots, v_n\}, \tag{1}$$

where $\mathcal{F}$ contains non-sequential user and context fields, such as *user ID, age group, membership level, device type, request time, and traffic source.* $\mathcal{S}$ is the chronological behavior sequence and includes *historical item-side features and action types, such as category, creator or seller, brand, and other available metadata.* $\mathcal{V}$ contains the current target-item features. During training, the latest behavior in each instance defines the target and feedback label, while $\mathcal{S}$ contains only preceding behaviors. After embedding and tokenization, $\boldsymbol{H}_{\mathcal{S}}$ and $\boldsymbol{H}_{\mathcal{F},\mathcal{V}}$ denote the sequence and joint non-sequential–target representation matrices. The following three paradigms progressively incorporate sequence representations into prediction.

2.1.1 ***Traditional Non-Sequential Modeling Paradigm.*** This paradigm models interactions between non-sequential heterogeneous features and target features:

$$\hat{y} = \phi_{\text{ns}}\big(\boldsymbol{H}_{\mathcal{F},\mathcal{V}}\big). \tag{2}$$

This formulation covers shallow and deep subnetworks that learn feature crosses from $\boldsymbol{H}_{\mathcal{F},\mathcal{V}}$ without modeling $\boldsymbol{H}_{\mathcal{S}}$. DeepFM [17] and DCNv2 [71] are representative examples. Because they omit temporal dependencies, UniRank treats them as conventional baselines outside the unified benchmark.

2.1.2 ***Stacked Unified Interaction.*** This paradigm first aggregates $\boldsymbol{H}_{\mathcal{S}}$ and then combines it with $\boldsymbol{H}_{\mathcal{F},\mathcal{V}}$:

$$\boldsymbol{H}_g = g_{\text{agg}}\big(\boldsymbol{H}_{\mathcal{S}}, \boldsymbol{H}_{\mathcal{F},\mathcal{V}}\big), \quad \hat{y} = \phi_{\text{stack}}\Big(\boldsymbol{H}_{\mathcal{F},\mathcal{V}}, \boldsymbol{H}_g\Big), \tag{3}$$

where $\boldsymbol{H}_g$ is the aggregated sequence representation matrix and ϕ_{stack} is the downstream interaction and prediction network. DIN [89] provides an early target-aware pooling example but lacks a token-based unified architecture and is excluded. UniRank evaluates token-based models such as HiFormer [16], RankMixer [92], and UniMixer [18].

2.1.3 ***Layer-wise Unified Interaction.*** This paradigm exchanges information between sequential and non-sequential representations within every interaction layer:

$$(\boldsymbol{H}_{\mathcal{S}}^{(l+1)}, \boldsymbol{H}_{\mathcal{F},\mathcal{V}}^{(l+1)}) = g^{(l)}(\boldsymbol{H}_{\mathcal{S}}^{(l)}, \boldsymbol{H}_{\mathcal{F},\mathcal{V}}^{(l)}), \quad \hat{y} = \phi_{\text{out}}(\boldsymbol{H}^{(L)}), \tag{4}$$

where $l \in \{0, \dots, L-1\}$ indexes the interaction layers, $\boldsymbol{H}_{\mathcal{S}}^{(l)}$ is the sequence representation matrix at layer l, and $\boldsymbol{H}_{\mathcal{F},\mathcal{V}}^{(l)}$ is the joint representation matrix of non-sequential and target tokens. The function $g^{(l)}$ performs unified interaction within layer l, $\boldsymbol{H}^{(L)}$ is the final representation matrix, and ϕ_{out} maps it to $\hat{y}$. Unlike stacked architectures, this formulation preserves behavior, field, and target representations across layers and supports iterative information exchange. OneTrans [85], TokenFormer [90], and UltraHSTU [11] instantiate this paradigm with distinct interaction operators.

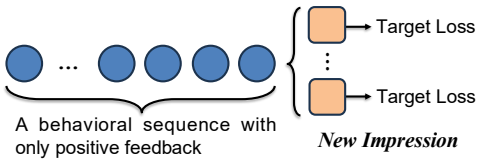


Figure 1: Traditional single-task New-Impression-Only paradigm.

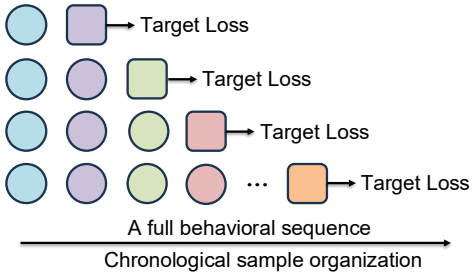


Figure 2: Pointwise autoregressive paradigm.

2.2 The Evolution of Training Paradigms

2.2.1 ***Traditional Single-Task New-Impression-Only Paradigm.*** Traditional ranking pipelines retain only positive behaviors in the user history and use this history to predict a new impression. Figure 1 treats the behavioral sequence as fixed auxiliary context: each new impression forms an independent sample and contributes one loss for a single feedback task, typically click prediction. The objective is

$$\mathcal{L}_{\text{NIO}} = \sum_{i=1}^{N} \text{BCE}\big(y_i, f_\theta(\mathcal{S}_i^+, \boldsymbol{x}_i)\big), \tag{5}$$

where N is the number of new-impression samples, $\mathcal{S}_i^+$ is the fixed positive-feedback sequence shared by multiple samples including i, $\boldsymbol{x}_i$ is the new-target feature vector, y_i is its binary label, and f_θ is the ranking model with parameters θ. For each sample i, Equation 5 applies the loss only to the new target $\boldsymbol{x}_i$. Events in $\mathcal{S}_i^+$ remain fixed auxiliary context and never become target positions, so they have neither labels nor loss terms and receive no direct per-position supervision. Consequently, gradients arise only from new targets, leaving sequences sparsely supervised.

2.2.2 ***Chronological Pointwise Autoregressive Paradigm.*** Figure 2 shows that UniRank constructs a chronological full-feedback sequence rather than a positive-only history, retaining positive actions and implicit negative outcomes such as exposed items without clicks [7]. Each eligible position t becomes an independent pointwise sample. The model predicts the current target's binary labels from the strictly preceding sequence $\mathcal{S}_{<t}$ of items, actions, and timestamps. Accordingly, a sample may activate multiple labels, and each model uses one prediction head per task. Its target loss sums binary cross-entropy terms:

$$\mathcal{L}_{\text{AR}} = \sum_{t \in \mathcal{T}} \sum_{k=1}^{K} \text{BCE}\big(y_{t,k}, f_\theta(\mathcal{S}_{<t}, \boldsymbol{x}_t)\big), \tag{6}$$

where $\mathcal{T}$ is the set of eligible target positions, K is the number of feedback tasks, $\mathcal{S}_{<t}$ is the full-feedback event sequence before position t, $\boldsymbol{x}_t$ is the current-target feature vector, and $y_{t,k}$ is its label for task k. Equation 6 provides up to $|\mathcal{T}|K$ supervised loss terms and direct gradients from successive positions rather than only the newest target. Because the chronological sequence retains every eligible event type, negative outcomes become both contextual representations and supervised targets. The resulting dense gradients improve the use of long interaction sequences and scale with chronological data and feedback tasks. This paradigm therefore better supports scaling in sequence length, data volume, and model capacity, while action-aware sequences capture dependencies across feedback tasks.

2.3 Representative Models

2.3.1 ***Stacked Unified Interaction.*** This paradigm arranges sequential modeling and feature interaction as consecutive modules. The sequence module first encodes the behavioral history, and the feature interaction module then combines this representation with user, target-item, and context features.

- **HiFormer** [16] pools the behavior history with target attention, tokenizes heterogeneous fields and task queries, and applies hierarchical HiFormer blocks to the stacked tokens.
- **RankMixer** [92] pools the history with target attention, tokenizes concatenated ranking fields into chunks, and alternates parameter-free Token Mixing with per-token MLPs.
- **Zenith** [84] separately embeds user and item identifiers, pools historical item attributes with target attention, tokenizes context and target attributes, and applies Zenith interaction blocks.
- **TokenMixer** [26] combines target-aware sequence summaries with global and group tokens, then applies reversible token mixing and per-token feed-forward networks.
- **UniMixer** [18] flattens histories with user and context fields, then chunk-tokenizes them for UniMixing-Lite blocks.
- **HeMix** [70] combines fixed and feature-conditioned queries with low-rank token mixing over recent and earlier histories.
- **SSR** [79] uses target-attention sequence pooling and multi-view competitive sparse filters with dense fusion.

2.3.2 ***Layer-wise Unified Interaction.*** This paradigm integrates sequential and non-sequential representations throughout every network layer.

- **OneTrans** [85] injects sequence and partitioned non-sequential tokens into every block, using mixed attention, a shared sequence FFN, and token-specific FFNs.
- **HyFormer** [24] encodes the sequence per layer, decodes global queries, and enriches them with non-sequential tokens.
- **MixFormer** [23] mixes non-sequential queries, uses cross-attention to retrieve information from the updated sequence, and fuses the resulting representations.
- **INFNet** [32] organizes categorical, sequential, and task tokens around a hub for layer-wise aggregation and broadcasting.
- **EST** [41] uses cross-attention to sequence representations and feature-specific feed-forward layers.
- **TokenFormer** [90] represents fields, histories, and targets as tokens with full interaction and sliding-window Flex Attention.
- **LONGER** [3] encodes local groups with inner Transformers and joins compressed long-term history with recent queries.

Table 1: Reproducibility requirement statistics.

Reproducibility requirements	HiFormer	RankMixer	Zenith	TokenMixer	UniMixer	HeMix	SSR	OneTrans	LONGER	HyFormer	MixFormer	INFNet	EST	TokenFormer	UltraHSTU	Ours
Unified data pipeline	△	△	△	△	△	△	△	△	△	△	△	△	△	△	△	✓
Unified task and metric definitions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	△	✓
Unified model source code	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Unified model hyperparameters	✓	△	△	△	✓	△	△	✓	✓	✓	△	✓	✓	△	✓	✓
Unified training pipeline	△	△	△	△	△	△	△	✓	△	△	△	△	△	△	△	✓
Unified efficiency optimization	△	△	△	△	△	△	△	✓	✓	△	△	△	△	△	✓	✓

- **UltraHSTU** [11] extends HSTU-style sequential transduction [82] with hybrid attention that combines sparse and full attention in each interaction layer.

3 Benchmarking Ranking Models

3.1 Reproducibility Analysis

Ranking models receive substantial attention from both academia and industry. Many advances are developed with proprietary data, implementations, and training infrastructure. Differences in access to these resources can make it difficult to determine whether gains arise from model architecture, data processing, training pipelines, or engineering optimization. Moreover, dataset and reproduction choices may alter model rankings even in controlled experiments. We therefore define six requirements for reproducible and fair ranking model research.

- **Unified data pipeline.** Studies use consistent raw-log filtering, feature extraction, categorical encoding, sequence construction, target alignment, chronological splitting, truncation, and padding. This creates matched samples and prevents future leakage from confounding comparisons.
- **Unified task and metric definitions.** Each feedback task uses consistent labels, sample eligibility rules, loss functions, metric implementations, and aggregation procedures. This ensures measured differences stem from architecture, not task formulation.
- **Unified model source code.** Baselines use complete, verifiable source code and consistent interfaces. This avoids confusing implementation differences with architecture.
- **Unified hyperparameters.** Shared capacity parameters such as network depth, token count, token dimension, embedding dimension, and sequence length should be aligned as closely as possible across baselines. This prevents capacity mismatches from driving apparent model gains, while architecture-specific parameters vary only for defining operations.
- **Unified training pipeline.** Baselines share data loading, training, validation, and testing. The pipeline also controls optimizers, learning rates, effective batch size, accumulation, initialization, epochs, and task weighting so that training choices do not masquerade as architectural improvements.
- **Unified efficiency optimization.** Studies align and disclose distributed training, precision, attention kernels, operator compilation, checkpointing, hardware, and software configurations. This makes performance and resource use comparable under the same systems setting.

Table 1 summarizes how existing studies cover the six requirements. The symbols ✓, △, and ✗ denote fully satisfied, unknown or partially satisfied, and not satisfied, respectively. Coverage varies across dimensions: task and metric definitions are often documented, whereas complete preprocessing pipelines, official implementations, capacity controls, training specifications, and efficiency settings are not always jointly available. Consequently, no existing study covers all six dimensions under a single public protocol. UniRank consolidates these requirements into a unified public benchmark for reproducible and fair comparison.

Table 2: Dataset statistics.

Dataset	#Instances	#Users	#Items	#Fields	#Tasks	#AvgL	#MaxL
QK-Video	493,306,303	4,996,176	3,752,235	10	4	99	6,013
KuaiRand	323,464,444	27,285	32,038,725	40	6	11,855	228,030
TAAC-25	757,207,146	7,706,778	15,707,425	30	2	98	100
Taobao	23,601,301	470,570	831,643	23	4	50	3,756
MerRec	172,304,959	1,697,072	42,577,610	20	5	102	26,576

3.2 Evaluation Protocol

3.2.1 ***Datasets.*** UniRank evaluates five large-scale public datasets spanning short-video recommendation, advertising, e-commerce, and C2C marketplaces. Table 2 summarizes their scale, feature fields, feedback tasks, and behavior lengths. The main benchmark fixes sequence length at 100 for controlled comparison across all 15 models. Dataset provenance, collection periods, raw interaction types, and task definitions are detailed in Appendix A.1.

3.2.2 ***Data Preprocessing.*** UniRank maps heterogeneous platform logs to a shared pointwise schema of *user*, *context*, *sequence*, and *action* features. Each retained interaction becomes a target sample, with its preceding item–action pairs forming the behavior sequence. The pipeline joins user and item side information, encodes categorical fields, maps infrequent values to OOV, and writes matched `data`, `user_info`, and `item_info` Parquet blocks for efficient loading. Dataset-specific construction and preprocessing rules are detailed in Appendix A.2.

3.2.3 ***Data Splitting.*** Figure 3 compares a randomized user-disjoint split with UniRank's chronological split. The user-disjoint protocol assigns each user and their entire interaction history exclusively to one data partition. In fact, this protocol is better suited to cold-start evaluation, as it primarily measures generalization to unseen users rather than future interactions from returning users. In addition, the partitions may cover overlapping calendar periods, allowing training interactions to occur after validation or test interactions from other users. Such temporal inversion exposes the model to future popularity patterns and platform trends, making offline evaluation less representative of deployment.

UniRank instead orders all interactions chronologically and partitions them into training, validation, and testing sets with an approximate 8:1:1 ratio. Within each user trajectory, each target interaction is constructed using only preceding behavior. The validation period supports model configuration tuning, while the test period evaluates performance on the latest interactions. This protocol

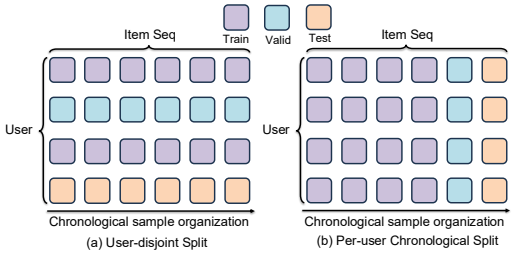


Figure 3: Comparison of randomized user-disjoint and per-user chronological sample splitting.

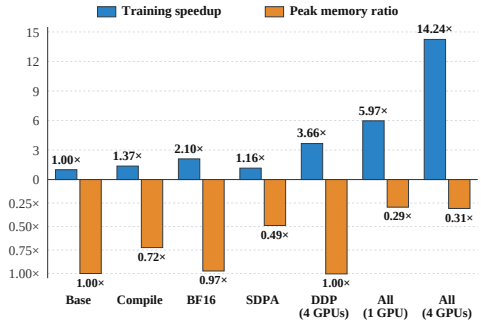


Figure 4: Training speedup and per-GPU peak memory.

emulates past-to-future prediction by training on users' earlier interactions and evaluating their subsequent interactions, while also assessing robustness to shifts in user interests over time.

3.2.4 ***Evaluation Metrics.*** The benchmark reports two widely used metrics, binary Logloss and AUC, for every task [25, 63, 93]. Lower Logloss indicates better probabilistic calibration, whereas higher AUC indicates better global discrimination.

3.2.5 ***Toolkit Implementation of UniRank.*** To support token-based ranking models with sequence lengths up to hundreds of thousands of items, UniRank provides an optimized PyTorch backend [54]. The toolkit includes the following components:

- **Distributed Data Parallel (DDP):** UniRank runs one CUDA process per GPU with NCCL gradient synchronization and partitions validation and testing across ranks.
- **Operator Compilation:** `torch.compile` uses the Inductor backend for eligible dense modules while leaving sparse embedding modules outside the compiled region.
- **Mixed Precision Training:** bf16 autocast reduces activation storage and enables Tensor Core compute, while binary cross-entropy loss is evaluated in FP32 for numerical stability.
- **Attention Optimization:** Models using scaled dot-product attention can dispatch to fused Flash Attention when PyTorch's dtype, shape, mask, and hardware constraints are satisfied. Flex Attention supports the structured sparse attention patterns used by TokenFormer and UltraHSTU.
- **Activation Checkpointing:** Ultra configurations use non-reentrant checkpointing for each main interaction block.
- **Blocked Data Pipeline:** Each split uses matching `data`, `user_info`, and `item_info` Parquet block triplets, block-local side-information caches, and per-rank load balancing.
- **Separate Dense and Sparse Optimization:** Dense parameters use AdamW and sparse embeddings use Adagrad with independent learning rates.

Figure 4 uses OneTrans with a sequence length of 1000 and normalizes throughput and per-GPU peak memory to the one-GPU base. With all optimizations enabled, All (4 GPUs) achieves a 14.24× speedup while reducing per-GPU peak memory by 69.2%, demonstrating the training and hardware efficiency of UniRank.

3.2.6 ***Training and Reproducibility.*** UniRank uses PyTorch [54] because its imperative execution, modular interfaces, and transparent debugging simplify development and reproduction. The default batch size is 8192 per GPU. If an experiment encounters an out-of-memory error, we reduce the per-GPU batch size linearly and increase gradient accumulation by the same factor to preserve the effective batch size. Each model trains for one epoch [86] with separate sparse and dense optimizers, consistent with production practice. Adagrad optimizes sparse embeddings at a default learning rate of 0.05, while AdamW optimizes dense parameters at 0.0001. All experiments run on NVIDIA H20 GPUs. The main benchmark fixes the embedding dimension at 16, network depth at three layers, and token dimension at 256. All interaction blocks use pre-norm for stable optimization. These controls align capacity and optimization budgets so that performance differences primarily reflect architectural choices. A grid search tunes model-specific hyperparameters. We release the training and evaluation code, preprocessing scripts, experiment configurations, and processed datasets in the GitHub repository, enabling end-to-end reconstruction.

3.3 Performance Analysis

Table 3 reports results on five datasets with a sequence length of 100. Bold values and parenthesized ranks identify the top three distinct four-decimal results for each task-metric pair. Ties share a rank, and subsequent ranks remain consecutive.

- **Both unified interaction paradigms are competitive:** Stacked models such as **HeMix** [70] and **UniMixer** [18], and layer-wise models such as **EST** [41] and **TokenFormer** [90], lead different datasets and tasks. Neither paradigm consistently dominates under the common protocol.
- **No universally superior model exists across data distributions:** Rankings vary substantially across QK-Video, KuaiRand, TAAC-25, Taobao, and MerRec. Platform-specific sequence patterns, feature distributions, and feedback frequencies materially affect performance, so findings may not transfer across datasets.
- **Model performance is highly dataset-dependent: EST** and **HeMix**, which originate from Taobao-oriented e-commerce ranking, are competitive on Taobao and MerRec. **TokenFormer** is competitive on the advertising dataset TAAC-25. Short-video and livestreaming models, including **UltraHSTU** [11], **UniMixer**, and **LONGER** [3], are competitive on QK-Video and KuaiRand. These patterns reflect production-specific inductive biases that transfer best across similar platforms.
- **No single model performs best across all tasks:** The strongest model changes across tasks within the same dataset. On MerRec, UniMixer, EST, and **TokenMixer** [26] lead like,

Table 3: Comprehensive Benchmarking Results (Sequence Length = 100)

Dataset	Task	Metric	HiFormer	RankMixer	Zenith	TokenMixer	UniMixer	HeMix	SSR	OneTrans	LONGER	HyFormer	MixFormer	INFNet	EST	TokenFormer	UltraHSTU
QK-Video	Click	AUC	0.9323	0.9318	0.9334	0.9336	0.9327	**0.9360 (2)**	0.9314	**0.9357 (3)**	0.9356	0.9330	0.9333	0.9332	0.9344	0.9350	**0.9363 (1)**
		Logloss	0.2539	0.2547	0.2521	0.2521	0.2533	**0.2478 (2)**	0.2554	**0.2484 (3)**	0.2486	0.2529	0.2525	0.2527	0.2505	0.2508	**0.2474 (1)**
	Follow	AUC	0.9185	0.9108	**0.9222 (2)**	**0.9219 (3)**	0.9023	**0.9227 (1)**	0.9081	0.9183	0.9166	0.9110	0.9173	0.9190	0.9208	0.9163	0.9217
		Logloss	**0.0050 (2)**	0.0052	**0.0050 (2)**	**0.0051 (3)**	0.0052	**0.0049 (1)**	**0.0051 (3)**	**0.0051 (3)**	**0.0051 (3)**	0.0052	**0.0051 (3)**	**0.0051 (3)**	**0.0050 (2)**	**0.0051 (3)**	**0.0049 (1)**
	Like	AUC	0.9465	0.9444	0.9480	0.9475	0.9428	**0.9496 (2)**	0.9428	0.9483	0.9489	0.9464	0.9470	0.9482	**0.9490 (3)**	0.9483	**0.9498 (1)**
		Logloss	0.0372	0.0375	0.0373	0.0374	0.0373	**0.0361 (1)**	0.0381	**0.0364 (3)**	**0.0362 (2)**	0.0370	0.0369	0.0370	0.0366	**0.0364 (3)**	**0.0361 (1)**
	Share	AUC	0.9334	0.9290	**0.9358 (2)**	0.9352	0.9239	**0.9368 (1)**	0.9292	0.9339	0.9330	0.9305	0.9320	0.9347	0.9345	0.9327	**0.9357 (3)**
		Logloss	**0.0053 (2)**	**0.0054 (3)**	**0.0053 (2)**	**0.0053 (2)**	0.0055	**0.0052 (1)**	0.0055	**0.0053 (2)**	**0.0053 (2)**	**0.0054 (3)**	0.0055	**0.0053 (2)**	**0.0053 (2)**	**0.0053 (2)**	**0.0053 (2)**
KuaiRand	Click	AUC	0.7782	0.7810	0.7816	0.7778	**0.7935 (3)**	0.7858	0.7872	0.7818	**0.7962 (2)**	0.7817	0.7795	0.7669	0.7719	**0.7990 (1)**	0.7744
		Logloss	0.5400	0.5408	0.5389	0.5440	**0.5253 (3)**	0.5351	0.5330	0.5531	**0.5226 (2)**	0.5412	0.5453	0.5677	0.5747	**0.5222 (1)**	0.5729
	Follow	AUC	**0.8872 (2)**	0.8739	0.8844	0.8865	0.8698	**0.8869 (3)**	0.8782	0.8715	0.8789	0.8800	0.8807	0.8728	**0.8921 (1)**	0.8762	0.8746
		Logloss	**0.0063 (3)**	0.0064	0.0065	**0.0062 (2)**	0.0064	**0.0061 (1)**	0.0064	0.0064	**0.0063 (3)**	0.0066	0.0066	0.0064	**0.0062 (2)**	**0.0062 (2)**	**0.0062 (2)**
	Like	AUC	**0.9294 (2)**	0.9241	**0.9294 (2)**	0.9270	0.9260	0.9266	0.9274	0.9215	**0.9281 (3)**	0.9215	0.9233	0.9207	**0.9301 (1)**	0.9193	0.9252
		Logloss	**0.0535 (1)**	0.0553	**0.0537 (2)**	0.0550	0.0543	0.0545	**0.0537 (2)**	0.0554	**0.0538 (3)**	0.0559	0.0551	0.0575	0.0545	0.0578	**0.0535 (1)**
	Comment	AUC	**0.8993 (2)**	0.8845	**0.8994 (1)**	**0.8993 (2)**	0.8870	**0.8974 (3)**	0.8886	0.8903	0.8928	0.8853	0.8894	0.8846	**0.8993 (2)**	0.8753	0.8949
		Logloss	**0.0147 (1)**	0.0152	**0.0147 (1)**	**0.0147 (1)**	**0.0150 (3)**	**0.0148 (2)**	**0.0150 (3)**	**0.0150 (3)**	**0.0150 (3)**	0.0151	**0.0150 (3)**	0.0155	**0.0148 (2)**	0.0155	**0.0148 (2)**
	Forward	AUC	0.8710	0.8499	0.8704	**0.8744 (2)**	0.8541	**0.8738 (3)**	0.8530	0.8622	0.8573	0.8691	0.8702	0.8592	**0.8788 (1)**	0.8537	0.8683
		Logloss	**0.0057 (2)**	**0.0058 (3)**	**0.0056 (1)**	**0.0056 (1)**	**0.0058 (3)**	**0.0057 (2)**	**0.0058 (3)**	0.0061	0.0059	**0.0058 (3)**	**0.0057 (2)**	0.0060	**0.0057 (2)**	**0.0058 (3)**	**0.0056 (1)**
	Long View	AUC	0.7961	0.8002	0.7981	0.7955	**0.8111 (3)**	0.7998	0.8056	0.8016	**0.8127 (2)**	0.7984	0.7982	0.7758	0.7935	**0.8157 (1)**	0.7985
		Logloss	0.4569	0.4539	0.4582	0.4600	**0.4420 (3)**	0.4589	0.4475	0.4597	**0.4405 (2)**	0.4552	0.4561	0.5048	0.4717	**0.4387 (1)**	0.4654
TAAC-25	Click	AUC	0.8310	0.8069	0.8236	0.8306	0.8089	0.8249	0.8113	0.8328	0.8335	0.8267	0.8132	0.8394	**0.8475 (2)**	**0.8532 (1)**	**0.8466 (3)**
		Logloss	**0.0355 (2)**	0.0412	0.0583	0.0393	0.0432	0.0645	0.0619	0.0439	0.0432	0.0527	0.0605	0.0501	**0.0335 (1)**	**0.0365 (3)**	0.0395
	Conversion	AUC	0.8730	0.8683	0.8627	0.8666	0.8626	0.8712	0.8716	0.8806	0.8863	0.8758	0.8754	**0.8866 (3)**	0.8622	**0.8871 (2)**	**0.8951 (1)**
		Logloss	**0.0177 (3)**	0.0216	0.0198	0.0191	0.0237	0.0241	0.0233	0.0232	0.0223	0.0185	0.0185	**0.0173 (2)**	**0.0157 (1)**	0.0182	0.0189
Taobao	Click	AUC	**0.6446 (3)**	0.6039	0.6217	0.6419	0.5999	**0.6523 (1)**	0.6006	0.6295	0.6377	0.6107	0.6444	0.6322	**0.6478 (2)**	0.6342	0.6366
		Logloss	0.1862	0.1887	0.1874	**0.1858 (3)**	0.1891	**0.1849 (1)**	0.1890	0.1886	0.1864	0.1890	**0.1857 (2)**	0.1887	0.1874	0.1867	0.1869
	Cart	AUC	0.8167	0.6754	0.7829	0.8253	0.7602	0.8284	0.7534	0.7458	0.8024	0.7484	0.8286	**0.8366 (2)**	**0.8404 (1)**	0.8217	**0.8333 (3)**
		Logloss	0.0183	0.0207	0.0192	0.0180	0.0192	**0.0178 (3)**	0.0201	0.0199	0.0188	0.0198	0.0182	0.0188	**0.0177 (2)**	**0.0178 (3)**	**0.0176 (1)**
	Favor	AUC	0.8471	0.7211	0.7993	0.8534	0.7957	0.8535	0.7816	0.7833	0.8224	0.7806	**0.8558 (3)**	**0.8597 (2)**	**0.8696 (1)**	0.8507	0.8544
		Logloss	0.0144	0.0155	0.0146	**0.0135 (3)**	0.0145	**0.0132 (1)**	0.0155	0.0148	0.0139	0.0150	**0.0135 (3)**	0.0137	**0.0132 (1)**	**0.0132 (1)**	**0.0133 (2)**
	Buy	AUC	0.8424	0.6856	0.7934	0.8497	0.7676	0.8533	0.7564	0.7673	0.8248	0.7485	0.8502	**0.8599 (2)**	**0.8684 (1)**	0.8483	**0.8562 (3)**
		Logloss	0.0069	0.0073	0.0068	**0.0064 (3)**	0.0069	**0.0061 (1)**	0.0072	0.0070	0.0070	0.0070	**0.0064 (3)**	0.0068	**0.0061 (1)**	**0.0062 (2)**	**0.0061 (1)**
MerRec	Like	AUC	0.7518	0.7448	0.7430	0.7533	**0.7650 (1)**	**0.7549 (2)**	0.7470	0.7490	0.7534	0.7361	0.7312	0.7296	**0.7543 (3)**	0.7434	0.7400
		Logloss	0.3277	0.3319	0.3306	0.3260	**0.3196 (1)**	**0.3256 (2)**	0.3296	0.3282	0.3278	0.3341	0.3433	0.3375	**0.3258 (3)**	0.3297	0.3343
	Cart	AUC	0.8123	0.7930	0.8019	**0.8133 (3)**	0.8115	**0.8136 (2)**	0.8005	0.8078	0.8081	0.7986	0.8012	0.7947	**0.8220 (1)**	0.8033	0.8012
		Logloss	0.0530	0.0539	0.0546	0.0526	**0.0523 (1)**	**0.0524 (2)**	0.0534	0.0531	**0.0525 (3)**	0.0538	0.0537	0.0542	**0.0523 (1)**	0.0534	0.0534
	Offer	AUC	0.7704	0.7288	0.7712	**0.7812 (1)**	0.7324	**0.7784 (2)**	0.7591	0.7518	0.7519	0.7400	0.7563	0.7378	**0.7724 (3)**	0.7390	0.7625
		Logloss	0.0242	0.0214	0.0221	**0.0210 (2)**	0.0223	**0.0208 (1)**	**0.0211 (3)**	0.0221	0.0213	0.0219	0.0212	0.0230	0.0224	0.0243	0.0212
	Checkout	AUC	0.8354	0.8379	0.8503	0.8562	0.8141	**0.8682 (1)**	0.8473	**0.8576 (3)**	**0.8623 (2)**	0.8514	0.8552	0.8447	0.8536	0.8305	0.8558
		Logloss	0.0127	**0.0047 (2)**	0.0127	0.0063	0.0068	**0.0046 (1)**	**0.0046 (1)**	0.0068	**0.0047 (2)**	0.0056	0.0055	0.0090	0.0121	0.0120	**0.0054 (3)**
	Purchase	AUC	0.8408	0.8407	0.8495	0.8593	0.8163	**0.8708 (1)**	0.8552	0.8591	**0.8653 (2)**	0.8475	0.8561	0.8465	**0.8600 (3)**	0.8335	0.8564
		Logloss	0.0077	**0.0030 (1)**	0.0086	0.0040	0.0044	**0.0030 (1)**	**0.0030 (1)**	0.0044	**0.0031 (2)**	0.0037	0.0037	0.0056	0.0082	0.0070	**0.0035 (3)**

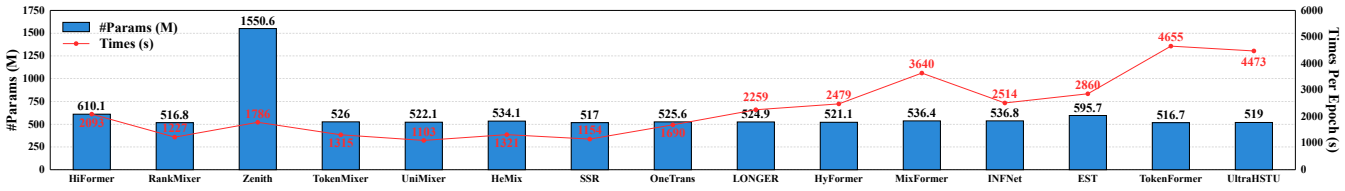


Figure 5: Efficiency comparisons on the KuaiRand dataset (Sequence Length = 100).

cart, and offer AUC, respectively, while HeMix leads checkout and purchase AUC. This motivates joint feedback evaluation.

3.4 Efficiency Analysis

Figure 5 compares total parameters and one-epoch training time under the same NVIDIA H20 GPUs, per-GPU batch size, sequence length, and precision policy. Zenith [84] has a substantially larger parameter footprint because it expands user and item ID embeddings and retains both IDs as independent tokens. Most of this capacity lies in sparse embedding tables, while target-aware pooling compresses the behavior sequence before interaction. Its training time therefore does not increase proportionally with parameter count. TokenFormer [90] and UltraHSTU [11] retain sequence tokens across interaction layers and apply structured sparse attention, which increases active computation. Flex Attention can further incur kernel-launch overhead on short sequences, so its training speed is not necessarily advantageous. Dedicated kernels and smaller sparse-attention windows may improve throughput. UniRank prioritizes fair model comparison and tunes each model by validation AUC rather than peak system throughput. Overall, parameter count reflects mostly static representation capacity, whereas epoch time reflects activated token computation.

Table 4: Model performance with different tokenization strategies

Dataset	Task	Metric	RankMixer				OneTrans			
			Chunk	Auto	Field	Random	Chunk	Auto	Field	Random
KuaiRand	Click	AUC	0.7810	**0.7848**	0.7628	0.7843	0.7574	0.7818	**0.7897**	0.7754
	Follow	AUC	**0.8739**	0.8709	0.8647	0.8674	0.8557	0.8715	**0.8839**	0.8828
	Like	AUC	0.9241	**0.9249**	0.9215	0.9248	0.8986	0.9215	**0.9243**	0.9242
	Comment	AUC	0.8845	**0.8911**	0.8613	0.8885	0.8743	0.8903	**0.8983**	0.8909
	Share	AUC	0.8499	**0.8590**	0.8501	0.8548	0.8583	0.8622	**0.8762**	0.8712
	Long View	AUC	0.8002	**0.8033**	0.7798	0.8032	0.7889	0.8016	**0.8049**	0.7897
	avg AUC	AUC	0.8523	**0.8557**	0.8400	0.8538	0.8389	0.8548	**0.8629**	0.8557
MerRec	Like	AUC	**0.7448**	0.7397	0.7429	0.7409	0.7451	0.7490	0.7441	**0.7493**
	Cart	AUC	0.7930	0.7879	**0.8031**	0.7924	0.8084	0.8078	0.8116	**0.8139**
	Offer	AUC	0.7288	0.7271	**0.7473**	0.7374	0.7610	0.7518	0.7549	**0.7728**
	Checkout	AUC	**0.8379**	0.8359	0.8378	0.8338	0.8607	0.8576	0.8560	**0.8713**
	Purchase	AUC	0.8407	0.8369	**0.8427**	0.8376	0.8624	0.8591	0.8523	**0.8748**
	avg AUC	AUC	0.7891	0.7855	**0.7948**	0.7884	0.8075	0.8051	0.8038	**0.8164**

Table 5: Model performance with different attention activations

Dataset	Task	Metric	OneTrans							
			Base	None	ReLU	SoftPlus	SiLU	GeLU	Sigmoid	Mish
KuaiRand	Click	AUC	0.7818	0.7951	0.7861	0.7859	**0.7978**	0.7968	0.7920	0.7961
	Follow	AUC	0.8715	0.8626	0.8712	0.8524	0.8722	**0.8753**	0.8731	0.8684
	Like	AUC	0.9215	0.9214	0.9242	0.9124	0.9231	0.9239	**0.9246**	0.9227
	Comment	AUC	**0.8903**	0.8861	0.8902	0.8662	0.8878	0.8896	0.8867	0.8894
	Share	AUC	**0.8622**	0.8519	0.8567	0.8536	0.8601	0.8603	0.8593	0.8569
	Long View	AUC	0.8016	0.8115	0.8038	0.8001	**0.8143**	0.8139	0.8111	0.8127
	avg AUC	AUC	0.8548	0.8548	0.8554	0.8451	0.8592	**0.8599**	0.8578	0.8577
MerRec	Like	AUC	0.7490	0.7507	0.7393	0.7510	0.7516	0.7535	**0.7582**	0.7473
	Cart	AUC	0.8078	0.8093	0.8086	0.8087	**0.8134**	0.8125	0.8087	0.8065
	Offer	AUC	0.7518	0.7617	0.7504	0.7438	0.7616	**0.7625**	0.7427	0.7578
	Checkout	AUC	0.8576	0.8586	0.8575	0.8587	**0.8593**	0.8588	0.8589	0.8592
	Buy	AUC	0.8591	0.8557	0.8586	**0.8620**	0.8566	0.8573	0.8618	0.8605
	avg AUC	AUC	0.8051	0.8072	0.8029	0.8048	0.8085	**0.8089**	0.8061	0.8063

Practical selection should therefore consider throughput, memory, storage, and sequence-length scaling together.

3.5 A Practical Handbook

3.5.1 ***Tokenization Strategies.***

- **Chunk** [92] flattens the fields, divides the resulting vector into contiguous chunks, and maps each chunk to one token.
- **Auto** [19] allows each token projection to access all features.
- **Field** [16] preserves field boundaries and independently maps each field to one token.
- **Random** [5] fixes a random field permutation, divides the fields into balanced groups, and maps each group to one token.

Table 4 shows that tokenization imposes an architectural inductive bias. Auto learns global cross-field dependencies, Field preserves semantic boundaries, and Chunk or Random promotes diverse token groups with higher informational entropy [62]. Their relative effectiveness depends on whether interaction layers favor global access, semantic isolation, or controlled grouping.

3.5.2 ***Attention Activations.*** We compare the Softmax baseline [68] with unnormalized logits [89], ReLU [78], SoftPlus [28], SiLU [12], GeLU [21], Sigmoid [28], and Mish [48]. Table 5 shows that smooth signed activations are the most robust. GeLU leads average AUC on both datasets and SiLU performs similarly, suggesting that signed responses outperform competitive or non-negative aggregation. SoftPlus degrades KuaiRand because weak positive weights accumulate noise. Task optima still vary: Sigmoid leads Like, whereas unnormalized logits benefit Click and Long View. Overall, GeLU and SiLU are strong performance-oriented defaults, while Softmax remains attractive when fused-kernel efficiency is important [9].

Table 6: Model performance with architectural enhancements

Dataset	Task	Metric	OneTrans							
			Base	SpecT	AttGate	RoPE	QKNorm	GNorm	AttRes	Tau
KuaiRand	Click	AUC	0.7818	0.7892	0.7946	0.7793	0.7963	**0.7972**	0.7823	0.7615
	Follow	AUC	0.8715	0.8835	0.8841	0.8747	0.8882	**0.8907**	0.8810	0.8753
	Like	AUC	0.9215	0.9258	**0.9295**	0.9226	0.9283	0.9292	0.9245	0.9200
	Comment	AUC	0.8903	0.8942	0.8969	0.8978	0.8975	**0.8982**	0.8952	0.8967
	Share	AUC	0.8622	0.8708	0.8646	0.8742	**0.8773**	0.8770	0.8691	0.8705
	Long View	AUC	0.8016	0.8044	0.8118	0.8017	**0.8120**	0.8084	0.8055	0.7918
	avg AUC	AUC	0.8548	0.8613	0.8636	0.8584	0.8666	**0.8668**	0.8596	0.8526
MerRec	Like	AUC	0.7490	0.7423	0.7534	**0.7647**	0.7445	0.7424	0.7575	0.7553
	Cart	AUC	0.8078	0.8082	0.8140	0.8077	**0.8192**	0.8113	0.8027	0.7909
	Offer	AUC	0.7518	0.7612	0.7591	**0.7642**	0.7520	0.7465	0.7567	0.7634
	Checkout	AUC	0.8576	0.8630	**0.8727**	0.8611	0.8622	0.8522	0.8617	0.8669
	Buy	AUC	0.8591	0.8644	**0.8758**	0.8645	0.8626	0.8565	0.8640	0.8677
	avg AUC	AUC	0.8051	0.8078	**0.8150**	0.8125	0.8081	0.8018	0.8085	0.8088

Table 7: Model performance with different optimizers

Dataset	Task	Metric	RankMixer					
			Base	Muon	LaProp	Scion	SOAP	RMSProp
KuaiRand	Click	AUC	0.7810	**0.7865**	0.7823	0.7778	0.7797	0.7824
	Follow	AUC	0.8739	0.8735	0.8767	0.8498	**0.8788**	0.8686
	Like	AUC	0.9241	0.9252	**0.9271**	0.9191	0.9262	0.9243
	Comment	AUC	0.8845	0.8849	0.8937	0.8728	**0.8944**	0.8826
	Share	AUC	0.8499	0.8578	**0.8715**	0.8066	0.8655	0.8543
	Long View	AUC	0.8002	**0.8024**	0.7994	0.7974	0.7979	0.8003
	avg AUC	AUC	0.8523	0.8551	**0.8584**	0.8373	0.8571	0.8521
MerRec	Like	AUC	0.7448	0.7592	0.7463	0.7461	**0.7619**	0.7480
	Cart	AUC	0.7930	0.8088	0.8023	0.8021	**0.8174**	0.7934
	Offer	AUC	0.7288	0.7630	0.7384	0.7336	**0.7800**	0.7265
	Checkout	AUC	0.8379	0.8311	0.8402	**0.8414**	0.8358	0.8369
	Buy	AUC	0.8407	0.8334	0.8465	0.8453	**0.8469**	0.8409
	avg AUC	AUC	0.7891	0.7991	0.7948	0.7937	**0.8084**	0.7892

3.5.3 ***Architectural Enhancements.***

- **SpecT** [76] adds learnable boundary tokens as attention sinks.
- **AttGate** [58] applies a learned sigmoid gate after attention.
- **RoPE** [65] rotates sequence queries and keys to encode position.
- **QKNorm** [22] normalizes attention queries and keys.
- **GNorm** [75] uses heterogeneous LayerNorm: sequence tokens share one, while non-sequential tokens use individual ones.
- **AttRes** [20] adds residual attention logits across adjacent layers.
- **Tau** learns a separate attention scale for each head.

Table 6 shows that these enhancements have substantially different cross-dataset robustness. AttGate is the only component that improves every task on both datasets. It therefore provides the safest general-purpose enhancement by suppressing target-irrelevant attention updates. QKNorm and GNorm perform best on KuaiRand, but GNorm degrades MerRec, suggesting that heterogeneous normalization may remove useful magnitude differences between token groups. RoPE provides larger gains on MerRec, indicating that chronological order has dataset-dependent importance. SpecT mainly benefits KuaiRand, whereas Tau reduces its average AUC. These results show that most enhancements align with dataset-specific characteristics rather than providing universal gains.

3.5.4 ***Dense Optimizers.*** We fix Adagrad for sparse embeddings and compare AdamW [44] with Muon [40], LaProp [43], Scion [56], SOAP [69], and RMSProp [67] for dense parameters. Table 7 shows that no optimizer consistently dominates across datasets. LaProp performs best on KuaiRand, with the largest gains on Comment and Share. SOAP performs best on MerRec, mainly

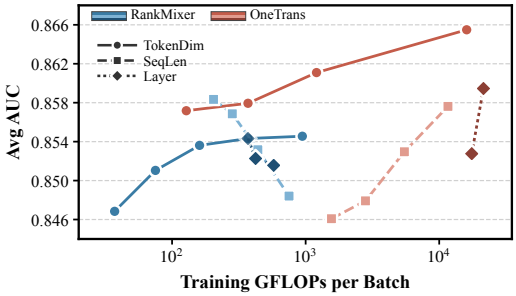


Figure 6: Scaling law study on KuaiRand.

through improvements on Cart, Offer, and Purchase. This difference suggests that optimizer effectiveness depends on dataset and task geometry. Muon provides moderate and relatively stable improvements, whereas RMSProp remains close to AdamW. Scion slightly improves MerRec but substantially degrades KuaiRand, showing that constrained updates can be distribution-sensitive. Overall, SOAP, Muon, and LaProp are relatively robust choices when accuracy is prioritized.

3.5.5 ***Scaling Laws.*** Figure 6 presents controlled scaling experiments on KuaiRand. Under identical data, task, and training settings, we vary token dimension, sequence length, and model depth, and compare average AUC with per-batch computation. RankMixer benefits primarily from wider token representations because its target-aware pooling compresses the behavior sequence before feature interaction, limiting the gains from longer sequences or additional layers. OneTrans progressively reduces active sequence tokens across layers while retaining non-sequential tokens. Its depth scaling must therefore be coupled with the initial sequence length. Short histories leave later layers too few sequence tokens to refine, whereas longer histories preserve sufficient information after progressive reduction. These results show that effective scaling should prioritize architecture-specific dimensions according to how sequential information is preserved, compressed, and propagated, rather than uniformly increasing width, sequence length, and depth. When no clear architectural preference is evident, increasing the token dimension provides a reliable default.

4 Related Work

Ranking Models. The development of ranking models follows four broad stages: shallow feature interaction, deep feature interaction, sequence modeling, and unified modeling. Early shallow methods assign weights to individual features or capture low-order interactions through factorization-based formulations [27, 51, 60, 61, 77]. Deep feature interaction methods subsequently embed sparse categorical features and replace manual feature crosses with neural interaction modules. Some studies learn implicit interactions through multilayer perceptrons or product transformations, while others explicitly construct high-order interactions through cross networks, self-attention, graph propagation, or polynomial functions [17, 30, 31, 35–37, 59, 64, 71, 72]. Further studies reduce manual design through automatic interaction selection, feature grouping, and architecture search [10, 38, 39, 47, 91]. Despite their effectiveness, these methods primarily treat each impression as an isolated set of non-sequential fields, making it difficult to capture the temporal evolution of user interests from behavior histories.

Sequence modeling methods address this limitation by aggregating behavior histories according to the target item and tracking the temporal evolution of user interest [13, 29, 89]. Multi-task methods further share information across clicks, conversions, and other feedback signals through shared representations, expert routing, and task-specific parameters [4, 45, 46, 52]. However, these systems commonly combine separate sequence modules, feature interaction modules, and task towers, resulting in heterogeneous architectures in which sequence modeling and feature interaction remain loosely coupled. Recent unified ranking methods instead represent behavior sequences, feature fields, targets, and task queries through a common token interface. Some studies perform sequence modeling before feature interaction [16, 18, 26, 70, 79, 84, 92], whereas others integrate both operations within each interaction layer [3, 11, 23, 24, 32, 41, 85, 90]. By replacing heterogeneous fine-grained modules with a common computational interface, this progression improves MFU and yields more favorable scaling curves as sequence length and model capacity increase [88, 92].

Benchmarking. Existing recommendation benchmarks provide complementary forms of standardization. FuxiCTR [25, 93] and DeepCTR [63] focus on single-task CTR models evaluated on conventional tabular datasets, while RecBole [87] supports a broader range of general and sequential recommendation. Scenario-Wise Rec [34] and MMLRec [80] further extend evaluation to multi-scenario or multi-task prediction, but offer limited support for industrial-scale behavioral sequences and token-based unified architectures. Overall, these benchmarks do not jointly standardize chronological sample construction, multi-task supervision, capacity-controlled model comparison, production-oriented training protocols, and efficiency optimization. To the best of our knowledge, UniRank is the first benchmark specifically designed for token-based unified ranking models. It integrates large-scale public datasets, long behavior sequences, multi-task supervision, controlled experimental configurations, and optimized training pipelines within a unified framework.

5 Conclusion and Future Work

This paper proposed UniRank, an open benchmark for ranking models with unified sequential modeling and feature interaction. UniRank provided standardized evaluation protocols, an optimized PyTorch toolkit, and five large-scale public datasets from diverse platforms. The evaluation covered over 700 million instances and sequences exceeding 10^5 interactions, while handbook experiments examined design choices under limited compute. Its chronological pointwise autoregressive paradigm and acceleration components supported reproducible comparison of modern ranking architectures and enabled practical studies of scaling laws. We reported results for 15 unified ranking models and established a foundation for more open research on modern recommender systems. Future versions will extend UniRank to multimodal ranking models and datasets such as TaoBao-MM [74] and MicroLens [50]. We will also add MerRec-1B [33], RecFlow [42], and VK-LSVD [57].

References

[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning A Deep Listwise Context Model for Ranking Refinement. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Ann Arbor, MI, USA, 135–144. doi:10.1145/3209978.3209985

[2] Alibaba. 2018. Taobao Display Ad Click-Log And User Behavior Log. Tianchi dataset.

[3] Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, Xionghang Xie, Shiru Ren, Xiang Sun, Yaocheng Tan, Peng Xu, Yuchao Zheng, and Di Wu. 2025. LONGER: Scaling up Long Sequence Modeling in Industrial Recommenders. *Proceedings of the Nineteenth ACM Conference on Recommender Systems* (2025). arXiv:2505.04421.

[4] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. Pepnet: Parameter And Embedding Personalized Network for Infusing with Personalized Prior Information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3795–3804.

[5] Jin Chen, Shangyu Zhang, Bin Hu, Chao Zhou, Junwei Pan, Gengsheng Xue, Wentao Ning, Gengyu Weng, Wang Zheng, Shaohua Liu, Zeen Xu, Chengyuan Mai, Shijie Quan, Tingyu Jiang, Lifeng Wang, Shudong Huang, Chengguo Yin, Haijie Gu, and Jie Jiang. 2026. RankUp: Towards High-Rank Representations for Large Scale Advertising Recommender Systems. *CoRR* abs/2604.17878 (2026). arXiv:2604.17878 doi:10.48550/ARXIV.2604.17878

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[7] Zexuan Cheng, Yue Liu, Jun Zhang, and Jie Jiang. 2026. Beyond Positive Signals: Unlocking Implicit Negative Behaviors for Enhanced Sequential User Modeling. *arXiv preprint arXiv:2606.15252* (2026).

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, Boston, MA, USA, 191–198. doi:10.1145/2959100.2959190

[9] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., Red Hook, NY, USA, 16344–16359.

[10] Wei Deng, Junwei Pan, Tian Zhou, Deguang Kong, Aaron Flores, and Guang Lin. 2021. Deeplight: Deep Lightweight Feature Interactions for Accelerating CTR Predictions in Ad Serving. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 922–930.

[11] Qin Ding, Kevin Course, Linjian Ma, Jianhui Sun, Ruochen Liu, Zhao Zhu, Chunxing Yin, Wei Li, Dai Li, Yu Shi, Xuan Cao, Ze Yang, Han Li, Xing Liu, Bi Xue, Hongwei Li, Rui Jian, Daisy Shi He, Jing Qian, Matt Ma, Qunshu Zhang, and Rui Li. 2026. Bending The Scaling Law Curve in Large-Scale Recommendation Systems. *arXiv preprint arXiv:2602.16986* (2026).

[12] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Networks* 107 (2018), 3–11.

[13] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-through Rate Prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[14] Chongming Gao, Shijun Li, Wenqiang Lei, et al. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.

[15] Siyu Gu and Xiangrong Sheng. 2022. On Ranking Consistency of Pre-Ranking Stage. *CoRR* abs/2205.01289 (2022). arXiv:2205.01289 doi:10.48550/arXiv.2205.01289

[16] Huan Gui, Ruoxi Wang, Ke Yin, Long Jin, Maciej Kula, Taibai Xu, Lichan Hong, and Ed H. Chi. 2023. Hiformer: Heterogeneous Feature Interactions Learning with Transformers for Recommender Systems. *arXiv preprint arXiv:2311.05884* (2023).

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 1725–1731.

[18] Mingming Ha, Guanchen Wang, Linxun Chen, Xuan Rao, Yuexin Shi, Tianbao Ma, Zhaojie Liu, Yunqian Fan, Zilong Lu, Yanan Niu, Han Li, and Kun Gai. 2026. UniMixer: A Unified Architecture for Scaling Laws in Recommendation Systems. *arXiv preprint arXiv:2604.00590* (2026).

[19] Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, Yueming Han, MengLei Zhou, Lei Yu, Chuan Liu, and Wei Lin. 2025. MTGR: Industrial-Scale Generative Recommendation Framework in Meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 5731–5738. doi:10.1145/3746252.3761565

[20] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. 2021. RealFormer: Transformer Likes Residual Attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 929–943. doi:10.18653/v1/2021.findings-acl.81

[21] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). arXiv:1606.08415 [cs.LG]

[22] Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. 2020. Query-Key Normalization for Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 4246–4253. doi:10.18653/v1/2020.findings-emnlp.379

[23] 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).

[24] Yunwen Huang, Shiyong Hong, Xijun Xiao, Jinqiu Jin, Xuanyuan Luo, Zhe Wang, Zheng Chai, Shikang Wu, Yuchao Zheng, and Jingjian Lin. 2026. HyFormer: Revisiting The Roles of Sequence Modeling And Feature Interaction in CTR Prediction. *arXiv preprint arXiv:2601.12681* (2026).

[25] Huawei. 2021. An Open-Source CTR Prediction Library. https://fuxictr.github.io.

[26] Yuchen Jiang, Jie Zhu, Xintian Han, Hui Lu, Kunmin Bai, Mingyu Yang, Shikang Wu, Ruihao Zhang, Wenlin Zhao, Shipeng Bai, Sijin Zhou, Huizhi Yang, Tianyi Liu, Wenda Liu, Ziyan Gong, Haoran Ding, Zheng Chai, Deping Xie, Zhe Chen, Yuchao Zheng, and Peng Xu. 2026. TokenMixer-Large: Scaling up Large Ranking Models in Industrial Recommenders. *arXiv preprint arXiv:2602.06563* (2026).

[27] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-Aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 43–50.

[28] Martin Kaloev and Georgi Krastev. 2021. Comparative Analysis of Activation Functions Used in The Hidden Layers of Deep Neural Networks. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 1–5.

[29] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, Beijing, China, 2615–2623. doi:10.1145/3357384.3357814

[30] Honghao Li, Yiwen Zhang, Yi Zhang, Hanwei Li, Lei Sang, and Jieming Zhu. 2026. FCN: Fusing Exponential And Linear Cross Network for Click-through Rate Prediction. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 681–691.

[31] Honghao Li, Yiwen Zhang, Yi Zhang, Lei Sang, and Jieming Zhu. 2025. Revisiting Feature Interactions from The Perspective of Quadratic Neural Networks for Click-through Rate Prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 1365–1375.

[32] Kaiyuan Li, Yongxiang Tang, Wenzheng Shu, Yanxiang Zeng, Chao Wang, Yanhua Cheng, Xialong Liu, and Peng Jiang. 2025. Aggregate And Broadcast: Scalable And Efficient Feature Interaction for Recommender Systems. *arXiv preprint arXiv:2508.11565* (2025).

[33] Lichi Li, Zainul Abi Din, Zhen Tan, Sam London, Tianlong Chen, and Ajay Daptardar. 2024. MerRec: A Large-Scale Multipurpose Mercari Dataset for Consumer-to-Consumer Recommendation Systems. *arXiv preprint arXiv:2402.14230* (2024).

[34] Xiaopeng Li, Jingtong Gao, Pengyue Jia, Xiangyu Zhao, Yichao Wang, Wanyu Wang, Yejing Wang, Yuhao Wang, Huifeng Guo, and Ruiming Tang. 2025. Scenario-Wise Rec: A Multi-Scenario Recommendation Benchmark. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. ACM, 1685–1695. doi:10.1145/3746252.3761137

[35] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. FiGNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 539–548.

[36] Zekun Li, Shu Wu, Zeyu Cui, and Xiaoyu Zhang. 2022. GraphFM: Graph Factorization Machines for Feature Interaction Modeling. *arXiv preprint arXiv:2105.11866* (2022).

[37] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit And Implicit Feature Interactions for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[38] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. AutoGroup: Automatic Feature Grouping for Modelling Explicit High-Order Feature Interactions in CTR Prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 199–208.

[39] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-through Rate Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2636–2645.

[40] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. 2025. Muon Is Scalable for LLM Training. arXiv:2502.16982 [cs.LG]

[41] Mingyang Liu, Yong Bai, Zhangming Chan, Sishuo Chen, Xiang-Rong Sheng, Han Zhu, Jian Xu, and Xinyang Chen. 2026. EST: Towards Efficient Scaling Laws in Click-through Rate Prediction via Unified Modeling. *arXiv preprint arXiv:2602.10811* (2026).

[42] Qi Liu, Kai Zheng, Rui Huang, Wuchao Li, Kuo Cai, Yuan Chai, Yanan Niu, Yiqun Hui, Bing Han, Na Mou, Hongning Wang, Wentian Bao, Yunen Yu, Guorui Zhou, Han Li, Yang Song, Defu Lian, and Kun Gai. 2025. RecFlow: An Industrial Full Flow Recommendation Dataset. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=vVHc8bGRns

[43] Ziyin Liu, Zhikang T. Wang, and Masahito Ueda. 2020. LaProp: Separating Momentum And Adaptivity in Adam. arXiv:2002.04839 [cs.LG]

[44] Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101* (2017).

[45] 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.

[46] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[47] Ze Meng, Jinnian Zhang, Yumeng Li, Jiancheng Li, Tanchao Zhu, and Lifeng Sun. 2021. A General Method for Automatic Discovery of Powerful Interactions in Click-through Rate Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1298–1307.

[48] Diganta Misra. 2019. Mish: A Self Regularized Non-Monotonic Activation Function. arXiv:1908.08681 [cs.LG]

[49] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization And Recommendation Systems. *arXiv preprint arXiv:1906.00091* (2019). https://arxiv.org/abs/1906.00091

[50] Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. 2023. A Content-Driven Micro-Video Recommendation Dataset at Scale. *arXiv preprint arXiv:2309.15379* (2023).

[51] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. Field-Weighted Factorization Machines for Click-through Rate Prediction in Display Advertising. In *Proceedings of the 2018 World Wide Web Conference*. 1349–1357.

[52] Junwei Pan, Wei Xue, Ximei Wang, Haibin Yu, Xun Liu, Shijie Quan, Xueming Qiu, Dapeng Liu, Lei Xiao, and Jie Jiang. 2024. Ads Recommendation in A Collapsed And Entangled World. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5566–5577.

[53] Junwei Pan, Wei Xue, Chao Zhou, Xing Zhou, Lunan Fan, Yanbo Wang, Haoran Xin, Zhiyu Hu, Yaozheng Wang, Fengye Xu, Yurong Yang, Xiaotian Li, Junbang Huo, Wentao Ning, Yuliang Sun, Chengguo Yin, Jun Zhang, Shudong Huang, Lei Xiao, Huan Yu, Irwin King, Haijie Gu, and Jie Jiang. 2026. Tencent Advertising Algorithm Challenge 2025: All-Modality Generative Recommendation. *arXiv preprint arXiv:2604.04976* (2026).

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32 (2019).

[55] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, and Dan Pei. 2019. Personalized Re-Ranking for Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, Copenhagen, Denmark, 3–11. doi:10.1145/3298689.3347000

[56] Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. 2025. Training Deep Learning Models with Norm-Constrained LMOs. arXiv:2502.07529 [cs.LG]

[57] Aleksandr Poslavsky, Alexander D'yakonov, Yuriy Dorn, and Andrey Zimovnov. 2026. VK-LSVD: A Large-Scale Industrial Dataset for Short-Video Recommendation. *arXiv preprint arXiv:2602.04567* (2026).

[58] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Gated Attention for Large Language Models: Non-Linearity, Sparsity, And Attention-Sink-Free. Advances in Neural Information Processing Systems 38. https://proceedings.neurips.cc/paper_files/paper/2025/hash/904e89bb4e632e75fb47f093b620b257-Abstract-Conference.html

[59] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-Based Neural Networks for User Response Prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[60] Steffen Rendle. 2010. Factorization Machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[61] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating The Click-through Rate for New Ads. In *Proceedings of the 16th International Conference on World Wide Web*. 521–530.

[62] Olivier Roy and Martin Vetterli. 2007. The Effective Rank: A Measure of Effective Dimensionality. In *Proceedings of the 15th European Signal Processing Conference*. IEEE, Poznan, Poland, 606–610.

[63] Weichen Shen. 2017. DeepCTR: Easy-to-Use, Modular And Extendible Package of Deep-Learning Based CTR Models. https://github.com/shenweichen/deepctr.

[64] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[65] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2024. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing* 568 (2024), 127063. doi:10.1016/j.neucom.2023.127063

[66] Yang Sun, Junwei Pan, Alex Zhang, and Aaron Flores. 2021. FM2: Field-Matrixed Factorization Machines for Recommender Systems. In *Proceedings of the Web Conference 2021*. 2828–2837.

[67] Tijmen Tieleman and Geoffrey Hinton. 2012. RMSProp: Divide The Gradient by A Running Average of Its Recent Magnitude. COURSERA: Neural Networks for Machine Learning, Lecture 6.5.

[68] 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).

[69] Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. 2025. SOAP: Improving And Stabilizing Shampoo Using Adam for Language Modeling. International Conference on Learning Representations. https://proceedings.iclr.cc/paper_files/paper/2025/hash/e988664070e9591f93fdcf605f7dc623-Abstract-Conference.html

[70] Fangye Wang, Guowei Yang, Xiaojiang Zhou, Song Yang, and Pengjie Wang. 2026. Query-Mixed Interest Extraction And Heterogeneous Interaction: A Scalable CTR Model for Industrial Recommender Systems. *arXiv preprint arXiv:2602.09387* (2026).

[71] 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.

[72] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCNv2: Improved Deep & Cross Network And Practical Lessons for Web-Scale Learning to Rank Systems. In *Proceedings of the Web Conference 2021*. 1785–1797.

[73] Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. COLD: Towards The Next Generation of Pre-Ranking System. *CoRR* abs/2007.16122 (2020). arXiv:2007.16122 https://arxiv.org/abs/2007.16122

[74] Bin Wu, Feifan Yang, Zhangming Chan, Yu-Ran Gu, Jiawei Feng, Chao Yi, Xiang-Rong Sheng, Han Zhu, Jian Xu, Mang Ye, and Bo Zheng. 2025. MUSE: A Simple Yet Effective Multimodal Search-Based Framework for Lifelong User Interest Modeling. arXiv:2512.07216 [cs.IR] https://arxiv.org/abs/2512.07216

[75] Yuxin Wu and Kaiming He. 2018. Group Normalization. In *Proceedings of the European Conference on Computer Vision*. Springer, Cham, 3–19. doi:10.1007/978-3-030-01261-8_1

[76] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient Streaming Language Models with Attention Sinks. International Conference on Learning Representations. arXiv:2309.17453 [cs.CL]

[77] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning The Weight of Feature Interactions via Attention Networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3119–3125.

[78] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv preprint arXiv:1505.00853* (2015).

[79] Yantao Yu, Sen Qiao, Lei Shen, Bing Wang, and Xiaoyi Zeng. 2026. Beyond Dense Connectivity: Explicit Sparsity for Scalable Recommendation. *arXiv preprint arXiv:2604.08011* (2026). Accepted at SIGIR 2026.

[80] Guanghu Yuan, Jieyu Yang, Shujie Li, Mingjie Zhong, Ang Li, Ke Ding, Yong He, Min Yang, Liang Zhang, Xiaolu Zhang, and Linjian Mo. 2024. MMLRec: A Unified Multi-Task And Multi-Scenario Learning Benchmark for Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 3063–3072. doi:10.1145/3627673.3679691

[81] Guanghu Yuan, Fajie Yuan, Yudong Li, Beibei Kong, Shujie Li, Lei Chen, Min Yang, Chenyun Yu, Bo Hu, Zang Li, Yu Xu, and Xiaohu Qie. 2022. TenRec: A Large-Scale Multipurpose Benchmark Dataset for Recommender Systems. *arXiv preprint arXiv:2210.10629* (2022).

[82] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. 2024. Actions Speak Louder Than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 58484–58509. https://proceedings.mlr.press/v235/zhai24a.html

[83] Buyun Zhang, Liang Luo, Xi Liu, Jay Li, Zeliang Chen, Weilin Zhang, Xiaohan Wei, Yuchen Hao, Michael Tsang, Wenjun Wang, et al. 2022. DHEN: A Deep And Hierarchical Ensemble Network for Large-Scale Click-through Rate Prediction. *arXiv preprint arXiv:2203.11014* (2022).

[84] Ruifeng Zhang, Zexi Huang, Zikai Wang, Ke Sun, Bohang Zheng, Yuchen Jiang, Zhe Chen, Zhen Ouyang, Huimin Xie, Phil Shen, Junlin Zhang, Yuchao Zheng, Wentao Guo, and Qinglei Wang. 2026. Zenith: Scaling up Ranking Models for Billion-Scale Livestreaming Recommendation. *arXiv preprint arXiv:2601.21285* (2026).

[85] 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* (Dubai, United Arab Emirates) *(WWW '26)*. Association for Computing Machinery, New York, NY, USA, 8162–8170. doi:10.1145/3774904.3792838

[86] Zhao-Yu Zhang, Xiang-Rong Sheng, Yujing Zhang, Biye Jiang, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. Towards Understanding The Overfitting Phenomenon of Deep Click-through Rate Models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2671–2680.

[87] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards A Unified, Comprehensive And Efficient Framework for Recommendation Algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, 4653–4664. doi:10.1145/3459637.3482016

[88] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. 2025. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* (2025).

[89] 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.

[90] Yifeng Zhou, Yuehong Hu, Zhixiang Feng, Junwei Pan, Kaihui Wu, Hanyong Li, Shangyu Zhang, Shudong Huang, Zhangbin Zhu, Chengguo Yin, Haijie Gu, and Jie Jiang. 2026. TokenFormer: Unify The Multi-Field And Sequential Recommendation Worlds. *arXiv preprint arXiv:2604.13737* (2026).

[91] Guanghui Zhu, Feng Cheng, Defu Lian, Chunfeng Yuan, and Yihua Huang. 2022. NAS-CTR: Efficient Neural Architecture Search for Click-through Rate Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 332–342.

[92] Jie Zhu, Zhifang Fan, Xiaoxie Zhu, Yuchen Jiang, Hangyu Wang, Xintian Han, Haoran Ding, Xinmin Wang, Wenlin Zhao, Zhen Gong, Huizhi Yang, Zheng Chai, Zhe Chen, Yuchao Zheng, Qiwei Chen, Feng Zhang, Xun Zhou, Peng Xu, Xiao Yang, Di Wu, and Zuotao Liu. 2025. RankMixer: Scaling up Ranking Models in Industrial Recommenders. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 6309–6316. doi:10.1145/3746252.3761507

[93] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-through Rate Prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2759–2769.

A Appendix

A.1 Dataset Descriptions

Table 8 inventories the source fields, ordering signals, and prediction targets used by UniRank. Processed examples retain global IDs, original user IDs, and available history length.

- **QK-Video** [81] is the short-video subset of TenRec from Tencent's recommendation feed on QQ Kandian, a news and short-video content platform. It contains ordered impressions from September 17 to December 7, 2021. Each row represents an exposed user–video pair. The four binary labels indicate whether the user clicks the video, follows its creator, likes it, or shares it. An all-zero row represents an exposure without any of these actions. The release also contains user gender and age, the number of watching behaviors on the video, and the video's content category. Its scale and strongly skewed user activity support multi-task ranking over heterogeneous histories. Because timestamps are unavailable, the source row order is the only temporal signal.
- **KuaiRand** [14] comes from Kuaishou, the short-video platform of Kuaishou Technology, and covers April 8 to May 8, 2022. It mixes ordinary recommendation traffic with random-intervention traffic, reducing policy-induced exposure bias. Each exposure includes millisecond time, request-tab context, play time, and six labels. `is_click` denotes a click in the two-column interface or a valid play in the single-column interface. `is_like`, `is_follow`, `is_comment`, and `is_forward` respectively denote liking, following the creator, commenting, and forwarding. `long_view` denotes completion for videos no longer than 18 seconds or at least 18 seconds of viewing for longer videos. Rich user profiles, video metadata, dense feedback, and exceptionally long histories make this dataset suitable for sequential and multi-task studies.
- **TAAC-25** [53] uses industrial traffic from Tencent's 2025 Advertising Algorithm Competition. The product surface and feature semantics are intentionally deidentified. The timestamps in the processed data span October 15, 2024 to August 5, 2025. Raw action type 0 denotes exposure, type 1 denotes an ad click, and type 2 denotes a downstream conversion whose business meaning is anonymized. We label both types 1 and 2 as click and label type 2 as conversion, so every conversion is also a click. Four scalar user fields, four list-valued user fields, and thirteen item fields retain their anonymous numeric IDs. The resulting click–conversion funnel and production-scale traffic provide the largest instance set in the benchmark.
- **Taobao** [2] comes from Alibaba Group's Taobao e-commerce and display-advertising platform. It links ad impressions, user profiles, ad features, and shopping behavior logs. Impressions span May 6–13, 2017, while the behavior log covers April 22–May 13, 2017. `is_click` indicates whether the user clicks an exposed ad. `cart`, `fav`, and `buy` indicate whether the user subsequently adds a matched product to the cart, favorites it, or purchases it. User fields describe demographic and consumption segments, whereas ad fields identify category, campaign, advertiser, brand, and price. The linked logs therefore capture both immediate ad response and delayed shopping intent.
- **MerRec** [33] comes from Mercari, Inc.'s consumer-to-consumer marketplace. The complete release contains timestamped events from May through October 2023. UniRank uses the largest October partition. Each sample is anchored at an item view. `Like` and `Cart` denote liking and adding that item to the cart, `Offer` denotes submitting a price offer, `Checkout` denotes starting the purchase flow, and `Purchase` denotes completing it. Product metadata includes a three-level category hierarchy, brand, condition, size, shipping party, color, and price. Unlike feed and advertising logs, its negative base is a retained view without the corresponding downstream action.

Table 8: Input fields and full-data label ratios normalized by the negative label.

Dataset	User fields	Context fields	Item and sequence fields	Label ratio
QK-Video	user_id gender age watching_times	Source impression order No timestamp No request context	item_id video_category preceding items four-bit action tokens	Exposure: 1× Click: 0.405× Follow: 0.00174× Like: 0.02098× Share: 0.00229×
KuaiRand	user_active_degree is_lowactive_period is_live_streamer is_video_author follow_user_num_range fans_user_num_range friend_user_num_range register_days_range onehot_feat0 onehot_feat1 onehot_feat2 onehot_feat3 onehot_feat4 onehot_feat5 onehot_feat6 onehot_feat7 onehot_feat8 onehot_feat9 onehot_feat10 onehot_feat11 onehot_feat12 onehot_feat13 onehot_feat14 onehot_feat15 onehot_feat16 onehot_feat17	tab day_of_week is_weekend hour time_ms (ordering)	video_id video_type primary_tag music_type duration_bucket preceding items six-bit actions	Exposure: 1× Click: 0.608× Follow: 0.00106× Like: 0.0169× Comment: 0.00251× Forward: 0.000871× Long view: 0.351×
TAAC-25	103 (scalar) 104 (scalar) 105 (scalar) 109 (scalar) 106 (list) 107 (list) 108 (list) 110 (list) lists capped at five values	day_of_week is_weekend hour timestamp (ordering)	100 101 102 112 114 115 116 117 118 119 120 121 122 preceding items exposure/click/conversion actions	Exposure: 1× Click: 0.0744× Conversion: 0.0336×
Taobao	cms_segid cms_group_id final_gender_code age_level pvalue_level shopping_level occupation new_user_class_level	pid is_weekend hour impression time (ordering)	adgroup_id cate_id campaign_id customer_id brand price_bucket preceding ads four-bit actions	Exposure: 1× Click: 0.0520× Cart: 0.00386× Favorite: 0.00284× Buy: 0.00150×
MerRec	user_id	session_id day_of_week is_weekend hour stime (ordering)	product_id price_bucket c0_id c1_id c2_id brand_id item_condition_id size_id shipper_id color preceding view anchors five-bit actions	View: 1× Like: 0.151× Cart: 0.0128× Offer: 0.00492× Checkout: 0.000871× Purchase: 0.000553×

The 1× entry denotes the baseline exposure sample, while 0.151×, for example, indicates that the corresponding action count is 15.1% of the baseline count.

Table 9: Dataset-specific sample construction, feature processing, and chronological splitting.

Dataset	Sample and label construction	Feature and action processing	Ordering and split
QK-Video	Retain each impression and pack its four binary labels into one history action.	Encode category, watching count, gender, and age with frequency-based OOV filtering.	Preserve per-user source order because date-based splitting is unavailable.
KuaiRand	Treat each exposure as a sample and pack six labels into one history action.	Derive calendar context, bucket video duration, and join user and video attributes.	Sort by millisecond time and use 26 training, 3 validation, and 3 test days.
TAAC-25	Remove all-negative users. Raw conversion events receive both conversion and click labels.	Cap list fields at five values and OOV-filter rare user and item values.	Use 226 training, 1 validation, and 68 test days with positive-label constraints.
Taobao	Anchor on ad impressions and attribute matched cart, favorite, and buy events within 24 hours.	Join user/ad features, derive time context, bucket price, and encode four-label actions.	Sort impressions by time and use 6 training, 1 validation, and 1 test day.
MerRec	Anchor on the first item view and attach later downstream events as labels.	Derive session/time context, bucket price, encode item fields, and pack five-label actions.	Sort view anchors by time and use 25 training, 3 validation, and 3 test days.

A.2 Dataset Preprocessing

All releases are converted to the same pointwise sequential format before training. By default, we remove users with fewer than ten eligible events and reserve 0 for padding and OOV. All datasets use `min_count=2` as the OOV threshold. When computing or memory resources are limited, this threshold can be increased to 10 to reduce vocabulary size and embedding-table memory. Global user and item IDs remain lossless, and the main benchmark fixes every feature embedding at 16 dimensions. Rather than treating raw magnitudes as continuous inputs, we discretize dataset-specific duration and price values and derive calendar context from timestamps. For label encoding, we follow ESMM [46] and represent feedback labels as a multi-hot vector, allowing one sample to activate multiple tasks. Each history event pairs its item with an action token mapped from the complete multi-hot pattern. The output consists of matched `data`, `user_info`, and `item_info` Parquet blocks, with complete item, action, and available timestamp sequences stored in `user_info`. Table 9 summarizes these rules.

- **QK-Video.** Each impression supplies four user fields, two item-related fields, and click, follow, like, and share labels. These binary labels form one multi-hot target, allowing concurrent feedback to remain positive. We map the 16 possible label combinations to history-action tokens. We keep separate global user and video vocabularies and place video category in `item_info`. Because the release has no timestamps, an ingestion index preserves per-user source order for the chronological split. A day-based split is unavailable.
- **KuaiRand.** We merge standard-policy and random-intervention traffic and join 26 user-profile fields with video type, first tag, music type, and duration. We convert millisecond timestamps to Asia/Shanghai date, weekday, weekend, and hour context. Video duration is discretized into seven intervals, from zero-length through longer than 60 seconds, and video metadata uses a separate categorical vocabulary. Click, follow, like, comment, forward, and long view form one multi-hot target, whose complete label combination maps to a history-action token. Chronological splitting uses 26 training days, 3 validation days, and 3 test days.
- **TAAC-25.** Each event contains four scalar and four list-valued user fields, thirteen item fields, and an anonymous action type. Type 0 is negative for both tasks, type 1 activates only click, and type 2 activates both click and conversion. This multi-hot organization preserves conversion as a subset of click. We remove users without any positive outcome and pad or truncate each list to five values. The shortest trailing interval containing both positive labels provides 68 test days, the shortest adjacent preceding interval provides 1 validation day, and the preceding 226 days form training.
- **Taobao.** Each ad impression anchors one example, and its `clk` value supplies the click label. We join eight user-profile fields and five ad attributes, then match cart, favorite, and purchase events from the same user, category, and brand within the following 24 hours. Click and the three downstream actions form one multi-hot target, and its complete label combination maps to a history-action token. Price is discretized into ten intervals, timestamps yield Beijing-time weekend and hour context, and chronological splitting uses 6 training days, 1 validation day, and 1 test day.
- **MerRec.** Events are sorted by user, timestamp, and source-row order, and only the earliest `item_view` for each user–item pair anchors a sample. Later like, cart, offer, checkout, and purchase events activate the corresponding entries in one multi-hot target but do not become separate samples or history tokens. The complete label combination maps to a history-action token. We retain session context and ten item fields, discretize price, and derive calendar fields. Chronological splitting uses 25 training days, 3 validation days, and 3 test days.