UxSID: Semantic-Aware User Interests Modeling for Ultra-Long Sequence

Hongwei Zhang[*,‡], **Qiqiang Zhong**[*], **Jiangxia Cao**[*], **Junfeng Shu**[†],
Yiyang Lv[‡], **Huanjie Wang**[‡], **Liwei Guan**[‡], **Jing Yao**, **Yiyu Wang**, **Zhaojie Liu**, **Han Li**
Kuaishou Technology
{zhanghongwei08,lifan11,caojiangxia,shujunfeng,lvyiyang,wanghuanjie,
guanliwei,yaojing03,liubin21,zhaotianxing,lihan08}@kuaishou.com

Abstract

Modeling ultra-long user behavior sequences is an essential task for capturing evolving preferences in modern recommender systems, and this research direction has contributed solid gains in the past several years. However, recommendation systems always serve enormous traffic, and extending user sequences sharply increases computational cost, which creates a difficult trade-off between efficiency and effectiveness. To scale to longer user sequences while keeping lightweight serving computation, existing works can be divided into two paradigms: (1) *Search-based Top-K selection*, which constructs an **item-specific** subsequence for each candidate item to avoid facing the ultra-long sequence directly; and (2) *pre-trained user-interest compression*, which maps an ultra-long user sequence into a small group of **item-agnostic** user interest memories so that the online model can perceive user long-term interests via this highly compressed dense memory. Besides the two technical routes (totally item-specific or item-agnostic), we argue that there exists an intermediate path not well explored: preserving partial relevance between the user sequence and the target item, while exposing only limited signals to guide the direction of interest compression. This design does not pursue item-specific user interest compression, but seeks semantic-group shared general user interest memory according to item attributes, where semantically similar items share the same compressed user interest memory. Motivated by this, we propose **UxSID**, a novel framework that bridges this gap by facilitating a target item semantic-aware interaction between **U**ser histories and candidate **S**emantic **ID**s (SIDs). Specifically, UxSID employs a dual-level attention strategy: it first extracts item-agnostic user interests from raw sequences, and then performs a semantic-specific query over global behaviors and those agnostic interests to generate semantic-specific preferences. By adopting this end-to-end architecture, UxSID generates offline embeddings that balance computational parsimony with target items' semantic awareness, while strictly preserving parity with online inference in constant time. Extensive public benchmarks and a large-scale A/B test demonstrate that UxSID achieves state-of-the-art performance, driving a 0.337% revenue lift in advertising.

1 Introduction

Massive industrial platforms like TikTok, Instagram Reels, and Kuaishou attract vast user bases and host millions of creators producing images, short videos, and live streams. To connect users and the platform's items, a powerful recommendation system (RecSys) is necessary to distribute the right items to the right users. Within this recommendation and consumption cycle, hundreds of millions

[*]Equal contribution. [†]Corresponding author. [‡]Work done during an internship at Kuaishou Technology.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

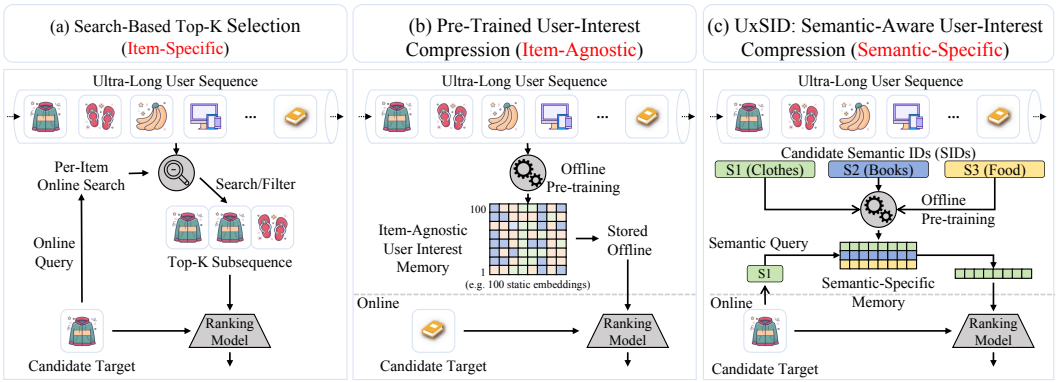


Figure 1: **Comparison of different paradigms for ULSM.** (a) **Item-specific Search**: Online filtering for each candidate, incurring high computational cost. (b) **Item-agnostic Compression**: Offline distillation into static memories, lacking target-specificity. (c) **UxSID**: A semantic-specific path that shares compressed interest memories among items with the same SIDs.

of users produce unprecedented volumes of daily behavioral data; for instance, active users often interact with as many as 10,000 items per week. As a result, Ultra-Long Sequence Modeling (ULSM) is paramount for modern recommender systems; it captures comprehensive preference evolution over extended horizons, bridging the gap between noisy short-term signals and underlying long-term user signals. By uncovering associations overlooked by truncated models, ULSM serves as a key driver for enhancing user engagement, conversion rates, and long-term loyalty [1–3].

Despite its importance, scaling sequence length introduces heavy computational burdens on industrial recommender systems that must handle massive real-time traffic. For instance, standard attention-based models [4] such as DIN [5], DIEN [6], and TransAct [7] are inherently constrained by the computational complexity; consequently, many systems only afford limited subsequence lengths of $n = 100$. As a result, many RecSys researchers are focusing on developing high-efficiency methods to scale user sequences with lightweight computation. Generally, there are two major ULSM technical roadmaps that have been well explored in recent years (a more comprehensive discussion of **Related Work** is deferred to Appendix A):

- Item-specific Top-K subsequence selection (Figure 1(a)). Methods such as SIM [8] and TWIN [9] utilize a Global Search Unit (GSU) to retrieve a Top-K subset of behaviors relevant to the target item prior to fine-grained modeling. Despite their efficiency, search-based methods suffer from inherent selection bias: whether via discrete hard matching or embedding-based soft retrieval, the filtering process is fundamentally constrained by the expressiveness of the pre-defined key space and stringent selection quotas.
- Item-agnostic compressed user interest memories (Figure 1(b)). Compression-based approaches, such as MIMN [2], LURM [10] and C-Former [11], aim to distill extensive histories into compact latent representations. However, these embeddings are typically target-agnostic, as they attempt to condense a user's diverse, evolving interests into static, user-centric embeddings [11, 12]. Consequently, these methods inevitably introduce irrelevant noise [13] and lack the target specificity required to disentangle a user's current intent from their vast historical background.

Besides the two technical routes (fully item-specific or entirely item-agnostic with target candidates), we argue that there exists an intermediate path not well explored: preserving partial relevance between the user sequence and the target item, while exposing only limited signals to guide the direction of interest compression. This design does not pursue "item-specific" user interest compression, but seeks semantic-group shared general user interest memory according to item attributes, where semantically similar items share the same compressed user interest memory, as shown in Figure 1(c).

To operationalize this intermediate path, we propose **UxSID** (Figure 2), a target semantic-aware compression framework that leverages Semantic IDs (SIDs) as the principled medium for interest distillation. Our core intuition is that SIDs, derived from deep quantization (e.g., RQ-VAE [14]), act as high-density semantic probes that naturally align with user interest clusters. While traditional IDs are discrete symbols, SIDs represent synthesized semantic clusters derived from multimodal content

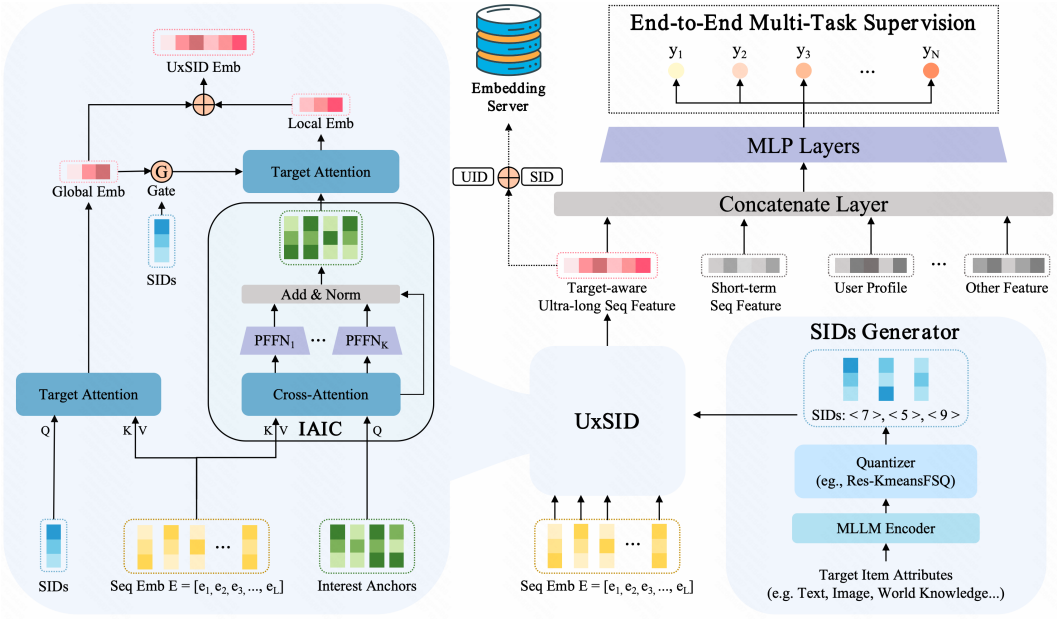


Figure 2: The architecture of UxSID primarily comprises three components: a target SIDs Generator that quantizes comprehensive attributes into semantically rich discrete IDs; a semantic-aware compression network featuring interest compression and hierarchical attention for sequence modeling; and an end-to-end multi-task supervision framework to ensure online-offline consistency.

[15, 16], collaborative signals [17–19], and world knowledge [20]. By utilizing SIDs to guide the compression process, UxSID can precisely extract target-relevant signals from expansive histories, rendering candidate-aware modeling both feasible and efficient for large-scale industrial deployment.

Specifically, we propose a hierarchical compression-and-probe strategy to ensure both global historical coverage and semantic-specific precision:

- **Item-Agnostic Interest Compression (IAIC).** This module condenses heterogeneous histories into a compact set of learnable interest anchors. By employing Per-token Feed-Forward Networks (PFFNs) and an orthogonality constraint, IAIC ensures each distilled interest captures a distinct facet of multifaceted preferences, effectively preserving representation diversity.
- **Hierarchical Semantic Probing.** It first extracts fine-grained global signals from the ultra-long sequence, then employs a gated semantic query to dynamically integrate information from the compressed interests. This process selectively amplifies target-relevant interests while filtering noise, ensuring the final representation encapsulates both immediate intent and long-term dependencies.

By integrating this strategy, UxSID generates compressed representations that maintain strict alignment between offline training and online inference. This architecture effectively overcomes the inherent selection bias of traditional paradigms, achieving target semantic-level resolution within expansive interaction histories. Our primary contributions are summarized as follows:

- **Target Semantic-Aware Behavior Compression.** To the best of our knowledge, UxSID is the first to enable SID-based semantic-specific compression in ULSM. By leveraging SIDs as queries, we shift the paradigm from static, user-centric distillation to target-aware interest compression.
- **Scalable UxSID Architecture.** We design a dual-stage framework featuring IAIC and Hierarchical Target Attention. This architecture ensures constant-time inference and stable resource consumption even as sequences scale to 10k interactions, effectively translating long-term dependencies into consistent ranking gains.
- **SOTA Performance and Industrial Impact.** We provide comprehensive validation on both public benchmarks and large-scale deployments at Kuaishou. UxSID consistently outperforms SOTA baselines and has delivered a 0.337% revenue increase in production, demonstrating its robust scalability and significant business value under stringent industrial latency constraints.

Paper Overview. Section 2 elaborates on the proposed methodology of UxSID and its industrial application pipeline. Section 3 presents extensive experimental results, spanning offline evaluations, a online A/B test, rigorous ablation studies, and case studies. Finally, Section 4 concludes the paper.

2 Methodology

In this section, we present the details of UxSID, a semantic-aware compression framework for ULSM. UxSID utilizes a hierarchical attention architecture to perform end-to-end interest compression and recommendation, ensuring high-fidelity interest perception.

Architecture Overview. The UxSID architecture comprises three core modules: (1) **SIDs Generation**, which leverages a Multimodal Large Language Model (MLLM) to transform heterogeneous item content into discrete semantic codes; (2) **Item-Agnostic Interest Compression**, which filters raw trajectories into structured interest anchors; and (3) **Hierarchical Semantic Probing**, which employs a gated dual-stage attention mechanism to resolve target-specific intent.

Semantic IDs Generation. To empower the model with a semantic probe that transcends literal ID matching, we generate SIDs via a reasoning-based alignment mechanism [20]. Given an item i with multimodal attributes (e.g., video frames and textual descriptions), we first employ an MLLM encoder to project it into a continuous business-aligned semantic space:

$$\mathbf{z}_i = \text{Enc}_{\text{MLLM}}(\text{Attributes}_i) \tag{1}$$

where $\mathbf{z}_i \in \mathbb{R}^d$. To ensure computational and storage efficiency, we apply a Res-KmeansFSQ hybrid quantization method [21]. This process decomposes $\mathbf{z}_i$ into M hierarchical levels:

$$\mathbf{z}_i \approx \sum_{m=1}^{M} \mathcal{C}_m(k_m), \quad k_m = \arg\min_j \|\mathbf{r}_{m-1} - \mathbf{c}_{m,j}\|_2 \tag{2}$$

where $\mathcal{C}_m$ represents the codebook at the m-th level, and $\mathbf{c}_{m,j} \in \mathbb{R}^d$ is the j-th codeword within that codebook. The vector $\mathbf{r}_m = \mathbf{z}_i - \sum_{l=1}^{m} \mathcal{C}_l(k_l)$ denotes the quantization residual at level m, with the initial residual defined as $\mathbf{r}_0 = \mathbf{z}_i$. Through this approach, the item is represented by a sequence of SIDs $(k_1, k_2, \ldots, k_M)$. In our industrial deployment, we primarily utilize the first-layer code k_1 as the target SIDs (c_{target}) to strike a balance between semantic granularity and inference latency.

Item-Agnostic Interest Compression. The IAIC block aims to condense the raw interaction sequence $\mathcal{B} = [b_1, b_2, \ldots, b_L]$ into a compact set of K interest anchors $\mathbf{P} \in \mathbb{R}^{K \times d}$ ($K \ll L$).

Interest Anchor Compression. We first transform each item i_t into a d-dimensional representation through an embedding lookup layer, resulting in the matrix $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L] \in \mathbb{R}^{L \times d}$. To extract representative signals, we define a set of learnable interest anchors $\mathbf{Q}_{anc} \in \mathbb{R}^{K \times d}$ that serve as interest queries to aggregate salient features via a cross-attention mechanism:

$$\mathbf{H} = \text{Softmax}\left(\frac{(\mathbf{Q}_{anc}\mathbf{W}^Q)(\mathbf{E}\mathbf{W}^K)^\top}{\sqrt{d}}\right)(\mathbf{E}\mathbf{W}^V) \tag{3}$$

where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]^\top \in \mathbb{R}^{K \times d}$ represents the compressed interest features, and $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ are learnable projection matrices. Besides, to enhance the independent representation of each interest anchor, we employ PFFNs. Specifically, for each anchor $\mathbf{h}_k$, we apply an anchor-specific sub-network followed by a residual connection and layer normalization:

$$\mathbf{p}_k = \text{LayerNorm}\left(\mathbf{h}_k + \sigma(\mathbf{h}_k\mathbf{W}_1^{(k)} + \mathbf{b}_1^{(k)})\mathbf{W}_2^{(k)} + \mathbf{b}_2^{(k)}\right) \tag{4}$$

where $\mathbf{W}_1^{(k)}, \mathbf{W}_2^{(k)}$ and $\mathbf{b}_1^{(k)}, \mathbf{b}_2^{(k)}$ are the learnable parameters specific to the k-th interest anchor, and $\sigma(\cdot)$ is the sigmoid function. This architecture ensures that the transformation is applied to each interest anchor independently, allowing them to be refined within their own semantic sub-spaces. The final set of compressed anchors is denoted as $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_K]^\top \in \mathbb{R}^{K \times d}$.

Diversity and Orthogonality Constraint. To ensure the interest anchors capture multifaceted preferences and avoid degenerate convergence toward a singular interest, we introduce a normalized Orthogonality Loss $\mathcal{L}_{ortho}$. By incorporating the squared L_2 norm of $\mathbf{P}$ in the denominator, we enforce structural independence between anchors while maintaining stability across varying scales:

$$\mathcal{L}_{ortho} = \left\| \frac{\mathbf{P}\mathbf{P}^T}{\|\mathbf{P}\|_2^2} - \mathbf{I} \right\|_F \tag{5}$$

where $\mathbf{I}$ is the identity matrix and $\|\cdot\|_F$ denotes the Frobenius norm. This constraint minimizes the redundancy between interest anchors, ensuring that each interest in the latent representation represents a distinct and well-separated facet of the user's long-term behavioral trajectory.

Hierarchical Semantic Probing. Unlike static methods, UxSID utilizes the target SIDs c_{target} as an active probe through a hierarchical dual-stage attention mechanism, refined by a gating module.

Explicit Semantic Probing. The first stage extracts fine-grained semantic signals by directly attending to the raw behavior sequence $\mathbf{E}$ using the target SIDs. This process captures global correlations between the target item and the user's expansive history:

$$\mathbf{e}_{global} = \text{Softmax}\left(\frac{(c_{target}\mathbf{W}_g^Q)(\mathbf{E}\mathbf{W}_g^K)^\top}{\sqrt{d}}\right)(\mathbf{E}\mathbf{W}_g^V) \tag{6}$$

where $\mathbf{e}_{global} \in \mathbb{R}^d$ represents the global interest response.

Gated Latent Probing. To refine interest resolution, the second stage conditions the probing process on the global context via a gating vector $\mathbf{g}_{ctx} \in \mathbb{R}^d$:

$$\mathbf{g}_{ctx} = \sigma\left(\text{GatedNet}(\mathbf{e}_{global})\right) \tag{7}$$

where $\text{GatedNet}(\cdot)$ is a two-layer MLP and $\sigma(\cdot)$ is the sigmoid function. This gating vector acts as a latent mask, modulating the target SID embedding c_{target} to produce a refined query $\mathbf{q}_{ref}$:

$$\mathbf{q}_{ref} = c_{target} \odot \mathbf{g}_{ctx} \tag{8}$$

By applying this Hadamard product, we align target semantic attributes with the user's global behaviors. Finally, the localized intent $\mathbf{e}_{local}$ is extracted from the interest anchors $\mathbf{P}$ via:

$$\mathbf{e}_{local} = \text{Softmax}\left(\frac{(\mathbf{q}_{ref}\mathbf{W}_l^Q)(\mathbf{P}\mathbf{W}_l^K)^\top}{\sqrt{d}}\right)(\mathbf{P}\mathbf{W}_l^V) \tag{9}$$

where $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d\times d}$ are learnable parameters.

The final target-aware representation is obtained by concatenating the dual-stage outputs: $\mathbf{E}^{\text{UxSID}} = [\mathbf{e}_{global}; \mathbf{e}_{local}]$. This hierarchical probing ensures that UxSID can simultaneously perceive both the broad historical context and the specific latent interest peaks relevant to the real-time recommendation.

Model Training and Loss Function. The precomputed $\mathbf{E}^{\text{UxSID}}$ is integrated with target features $\mathbf{E}^t$, user profiles $\mathbf{E}^u$, context $\mathbf{E}^c$, and short-term behaviors $\mathbf{E}^{\text{short}}$. The prediction $p(x)$ is formalized as:

$$p(x) = \sigma\left(\text{MLP}\left(\mathbf{E}^t; \mathbf{E}^u; \mathbf{E}^c; \mathbf{E}^{\text{short}}; \mathbf{E}^{\text{UxSID}} \mid x\right)\right) \tag{10}$$

The model is optimized end-to-end via a joint loss function:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log(p(x_n)) + (1-y_n)\log(1-p(x_n))] + \lambda\mathcal{L}_{ortho} \tag{11}$$

where the first term is the Binary Cross-Entropy loss for the recommendation task.

Serving in Production. The offline pipeline ensures the final representation $\mathbf{E}^{\text{UxSID}}$ is target-aware and task-aligned. After training, $\mathbf{E}^{\text{UxSID}}$ is precomputed and cached in Embedding Server (ES). The storage key is generated via a bitwise concatenation of user ID (UID) and target item SID (SID):

$$\text{Key} = \text{Hash}(UID \oplus SID), \quad \text{Value} = \mathbf{E}^{\text{UxSID}} \tag{12}$$

In the real-time serving stage, the system performs an $O(1)$ point lookup to retrieve $\mathbf{E}^{\text{UxSID}}$ from ES. The strong representational capacity of SIDs allows for a limited and manageable set of unique IDs per user; thus, the total storage footprint remains well within industrial feasibility (see Appendix B). The compressed embeddings then function as target-aware user history keys/values, which interact with the target item query (augmented with side information) via a lightweight attention mechanism for final ranking, thereby effectively meeting stringent industrial latency requirements.

3 Experiments

Datasets. The effectiveness of UxSID is evaluated on two public benchmarks, XLong [22] and KuaiRec-Big [23], alongside a large-scale industrial dataset. These public benchmarks were selected because they provide paired content features, which are indispensable for the training and semantic alignment of SIDs. Detailed statistics and preprocessing procedures are provided in Appendix C.

Baselines. To evaluate the effectiveness of UxSID, we compare it against several competitive baselines (which are detailed in Appendix D): DIN [5], SIM [8], ETA [24], SDIM [25], MIRRN [26], TWIN [9], C-Former [11]. All baselines share the bottom-level layers, varying only in ULSM.

Metrics. We evaluate model performance using AUC, UAUC, and WUAUC to measure overall and user-level ranking performance, referencing studies like [9, 20, 27]. Furthermore, we introduce Interest Recall@K (Int.R@K) in ablation and case analysis studies to quantify semantic activation precision, defined as the proportion of the Top-K behaviors (retrieved via attention score during explicit semantic probing) that share the same first-layer SIDs or category (tag) with the target item.

Implementation Details. We followed the experimental settings of [11, 26], including data partitioning and hyperparameter configurations. Baseline evaluations use sequence lengths of 2k for KuaiRec-Big and 1k for XLong, except for DIN which uses 100. The GSU retrieval quantity is also set to 100. The prediction heads of all models consist of an MLP with $\{200, 80, 2\}$ hidden size, and the sparse embedding dimension is fixed at 16. Notably, main evaluations are conducted with 1k length for industrial dataset, while scalability up to 10k is specifically analyzed in Section 3.3. All public experiments are trained using the Adam optimizer with a batch size of 256 and a learning rate of 0.001. Specifically for UxSID, the gating network is configured as $\{16, 16\}$ with an activation layer, while each FFN in the PFFNs follows a $\{16, 32, 16\}$ structure. We adopted LETTER [28] to construct the codebook and train the SIDs with a shape of $\{256, 256, 256, 256\}$ for public datasets. Ultimately, the first-layer SIDs were utilized, and the number of IAIC anchors is 16.

3.1 Overall Performance

Performance on Public Datasets. We first conduct experiments on two widely-recognized public benchmarks: XLong and KuaiRec-Big. As illustrated in Table 1, UxSID consistently achieves SOTA performance, outperforming all baseline paradigms. Furthermore, we show in Table 7 (Appendix E.1) that the standard deviation in the metrics for UxSID is insignificant.

Table 1: AUC Performance comparison with baselines. The best results are in bold. '-' indicates category unavailability.

Models	XLong	KuaiRec-Big
DIN	0.7889	0.8181
SIM-Hard	–	0.8201
SIM-Soft	0.7971	0.8279
ETA	0.7910	0.8231
SDIM	0.7915	0.8209
MIRRN	0.7926	0.8217
TWIN	0.8154	0.8269
C-Former	0.8135	0.8276
UxSID	**0.8408**	**0.8348**

Specifically, on XLong, UxSID achieves 0.8408 AUC, surpassing the strongest search-based baseline (TWIN) and the advanced compression model (C-Former). The gain over C-Former is noteworthy: while C-Former leverages learnable anchors for clustering, it remains inherently target-agnostic during compression. In contrast, UxSID utilizes SIDs as semantic queries to activate target-specific interests, demonstrating that target-aware perception is critical for resolving fine-grained preferences in ULSM.

Performance on Large-Scale Industrial Datasets. We further evaluate UxSID on a massive industrial dataset from Kuaishou, where a 0.1% improvement represents a significant milestone.

As reported in Table 2, UxSID achieves a CTCVR AUC of 0.8626, significantly outperforming the strongest industrial baselines, including SIM-Soft (+0.18%) and TWIN (+0.17%). This superior performance is primarily attributed to our IAIC block and the hierarchical probe architecture. While SIM and TWIN possess certain item-specific capabilities via retrieval or attention, UxSID further extends this advantage by leveraging the high-density semantics of SIDs to navigate the expansive interest landscape. Unlike traditional heuristic-based filtering which leads to information loss, UxSID's superior structure maintains global interest-capturing capabilities by precisely bridging the gap between dense user histories and target semantic intents.

Table 2: Performance comparison between UxSID and other baseline models.

Models	CTR			CTCVR		
	AUC	UAUC	WUAUC	AUC	UAUC	WUAUC
SIM-Hard	0.8698	0.6042	0.6063	0.8599	0.6161	0.6221
SIM-Soft	0.8711	0.6084	0.6099	0.8608	0.6228	0.6307
TWIN	0.8712	0.6093	0.6104	0.8609	0.6232	0.6310
UxSID (Ours)	**0.8728**	**0.6125**	**0.6161**	**0.8626**	**0.6269**	**0.6350**

Performance in Online A/B Testing. We deployed UxSID on Kuaishou's short-video advertising platform, where a week-long online A/B test (Table 3) demonstrated its effectiveness through significant performance gains.

Table 3: Online A/B results.

Scenarios	Advertising Metrics		
	Exposure	Cost	Revenue
Advertising	+0.111%	+0.231%	+0.337%

The significant divergence between Revenue (+0.337%) and Exposure (+0.111%) is particularly revealing, highlighting a shift towards high-precision conversion. These results demonstrate that, through semantic-specific compression and storage, UxSID renders end-to-end ULSM not only highly accurate but also computationally viable for industrial-level traffic under stringent latency constraints (details in Appendix B).

3.2 Ablation Study

Table 4: UxSID ablation results. '-' indicates ablation-induced or feature unavailability.

Variants	Industrial Dataset							Public Datasets		
	CTR			CTCVR				XLong	KuaiRec-Big	
	AUC	UAUC	WUAUC	AUC	UAUC	WUAUC	Int.R@50	AUC	AUC	Int.R@50
Category (Tag)	0.8707	0.6081	0.6088	0.8605	0.6186	0.6288	0.0543	-	0.8261	0.0916
w/o $\mathbf{e}_{global}$	0.8714	0.6101	0.6108	0.8615	0.6230	0.6318	-	0.8370	0.8302	-
w/o $\mathbf{e}_{local}$	0.8719	0.6114	0.6121	0.8618	0.6238	0.6327	0.1454	0.8375	0.8314	0.2009
w/o $\mathcal{L}_{ortho}$	0.8725	0.6119	0.6144	0.8624	0.6261	0.6342	0.1471	0.8385	0.8344	0.2063
w/o Gate	0.8723	0.6116	0.6127	0.8623	0.6249	0.6334	0.1467	0.8376	0.8342	0.2044
UxSID	**0.8728**	**0.6125**	**0.6161**	**0.8626**	**0.6269**	**0.6350**	**0.1488**	**0.8408**	**0.8348**	**0.2071**

We perform a comprehensive ablation study on UxSID to assess its components and target semantic-specific perception (Table 4).

Effectiveness of SID-based Semantic Querying. We evaluate query granularity by replacing candidate SIDs with coarse-grained category (tag) attributes. As shown in Table 4, SIDs consistently outperform tag-based queries across all metrics. This confirms that category-level attributes lack the semantic resolution required to precisely navigate complex user interest distributions.

Conversely, SIDs act as high-density semantic probes, enabling granular activation of historical interests. As evidenced by the Int.R@50 metric, this probing mechanism enhances the model's ability to identify historical interactions within the target's semantic cluster. This increased recall likely drives the observed AUC gains, empirically demonstrating that UxSID successfully bridges the target-agnostic bottleneck inherent in traditional compression paradigms.

Impact of Hierarchical Semantic Probing. Removing either $\mathbf{e}_{global}$ or $\mathbf{e}_{local}$ leads to noticeable performance degradation. Specifically:

- **Compressed Global Signals (*w/o* $\mathbf{e}_{global}$)**: Eliminating the explicit attention results in a significant drop in AUC. This indicates that fine-grained, item-to-item semantic signals are indispensable for capturing immediate target-specific relevance that might be smoothed over during compression.

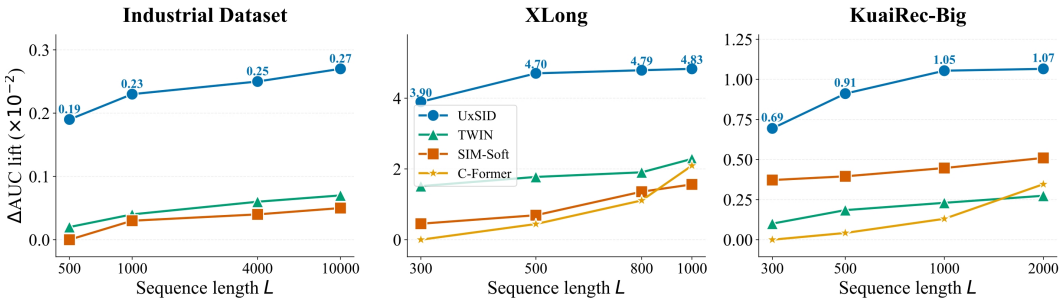


Figure 3: AUC improvements (percentage points) across various sequence lengths on all datasets.

- **Compressed Local Interests (*w/o* $\mathbf{e}_{local}$)**: The observed decline suggests that interest anchors are crucial for filtering historical noise and providing a diverse, structural view of user preferences that $\mathbf{e}_{global}$ cannot effectively capture.

Role of Gated Probing and Diversity Loss. Ablating GatedNet$(\cdot)$ reveals that direct probing with original SIDs is suboptimal. The gating mechanism ensures the local latent probe is precisely conditioned on the global context from the first stage, enhancing robustness during querying. Furthermore, removing $\mathcal{L}_{ortho}$ adversely affects performance, validating that forcing distinct interest clusters prevents mode collapse and preserves the multi-faceted nature of user preferences.

Appendix E.2 confirms UxSID's gains stem from its architecture rather than SIDs injection.

3.3 Efficiency and Scaling Law Analysis

Theoretical complexities are summarized in Table 5. Search-based paradigms (SIM, TWIN) suffer from online matching overhead. Conversely, UxSID offloads ULSM offline, achieving $O(1)$ online complexity for target-level compression and maintaining constant latency even when scaling to 10k.

Figure 3 illustrates the performance trends as sequence length scales, revealing two primary findings:

Limitations of Retrieval and Static Compression. As sequence lengths increase, search-based models (SIM and TWIN) exhibit decelerating growth; their fixed retrieval scope inevitably excludes distal but relevant interactions. Simultaneously, C-Former's performance is unstable on short sequences but scales as histories lengthen. This highlights a flaw in static compression: without target-aware guidance, the model struggles to disentangle signals from mounting noise, failing to maintain robust representations across sequence scales.

Table 5: Inference time complexity comparison. B: batch size, L: original sequence length, R: retrieved sequence length, d: hidden size, A: attribute index size, m: hash functions, c: compressed interest length, f: feature numbers.

Model	Inference Time Complexity
SIM-Hard	$(B\log(A) + BRd)$
SIM-Soft	$(BLd + BRd)$
ETA	$(BLm + BRd)$
SDIM	$Bm\log(d)$
MIRRN	$BLm + BR\log(R)d + BRd^2$
TWIN	$(BL + BfLd + BRd)$
C-Former	(BRd)
UxSID	(Bcd)

Scalability and Efficiency. UxSID consistently maintains the highest AUC, with the performance gap widening at the 10k scale. This confirms that SID-based routing effectively pinpoints specific interests, translating expansive behavioral data into substantial gains. Notably, this is achieved with constant online consumption, demonstrating UxSID's potential for modeling lifelong behaviors.

3.4 Storage Feasibility Analysis

Deployability of this lookup depends on how many keys the cache holds. User-level compression stores one group of tokens per user, while UxSID stores one compressed representation per active SID of that user and thus has a larger key space in form. The key set stays small in practice, because a user's interests concentrate in a limited number of SID buckets. Each cached value is the

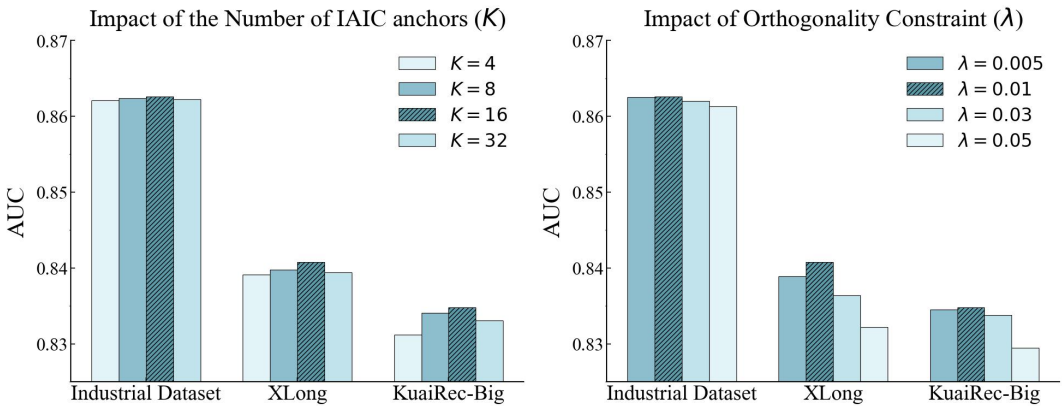


Figure 4: Hyperparameters analysis of UxSID on all three datasets.

representation of shape $[2, 32]$, at $s = 128$ bytes per key. On a platform with $|U| = 4 \times 10^8$ daily active users and about 50 active SIDs per user monthly, the cache holds $N_{\text{keys}} \approx 2 \times 10^{10}$ keys and occupies $\text{Mem} \approx 2.5$ TB. A few terabytes lie within the capacity of a distributed key-value store, so Eq. (12) is a constant-time read over a bounded key set. The same count is capped by the codebook: n_{sid} tracks how many semantic buckets a user has touched, and a user's interests are finite, so the working set stays near this scale however long the history grows. An item-specific cache has no such cap. Its size is $|U|$ times the number of distinct videos each user has watched, and it grows linearly with that viewing count. At the traffic of large industrial platforms, the resulting volume exceeds the petabyte scale, varying with the scenario, and it keeps rising as users watch more videos. UxSID does not have this problem in real storage.

Cache refresh and sequence length leave this storage budget unchanged, and it grows only when the user watches a video whose SID the user has not seen. An already-seen SID overwrites the same entry and allocates no new key, so existing keys stay unchanged. History length behaves the same way. The number of compressed tokens is a fixed hyper-parameter: users with different sequence lengths produce interest representations of the same size, so both the storage size and the online cost are invariant to L. Only the offline compression compute grows with L. The longer histories in Figure 3 are therefore absorbed at no extra storage and no extra serving cost.

3.5 Parameter Sensitivity Study

We conduct a sensitivity analysis across all datasets, with results summarized in Figure 4.

Impact of the Number of IAIC anchors (K). K dictates the capacity of the IAIC block to capture diverse intents. At $K = 4$, performance drops, suggesting that compressing ultra-long behavior into a highly bottlenecked embedding forces interest entanglement and the loss of fine-grained information. Performance improves as K grows, peaking at 16. However, excessively large K introduces redundancy and noise, causing overly dispersed routing and slightly degraded metrics.

Impact of Orthogonality Constraint (λ). λ controls the intensity of the diversity regularization applied to the compressed interests. Without sufficient penalty, the interest anchors tend to capture high-frequency patterns, reducing the diversity of the IAIC block. However, an excessively large λ overly constrains the latent space, forcing separation that disrupts the primary CTR prediction.

3.6 Visualization and Case Study

We investigated the mechanisms underlying UxSID's efficacy, focusing on candidate SID activation within sequences and compressed anchors on KuaiRec-Big. As shown in Figure 5(a), distinct attention routing patterns for different candidates demonstrate that SIDs successfully trigger diverse semantic interests. The contrast in Int.R@50 between positive and negative samples validates the alignment between activation relevance and CTR objectives: negative items yield low Int.R@50 due to semantic irrelevance, whereas positive samples achieve significantly higher recall. Notably, UxSID activates data across the entire behavioral cycle; even early behaviors are assigned high attention weights, a

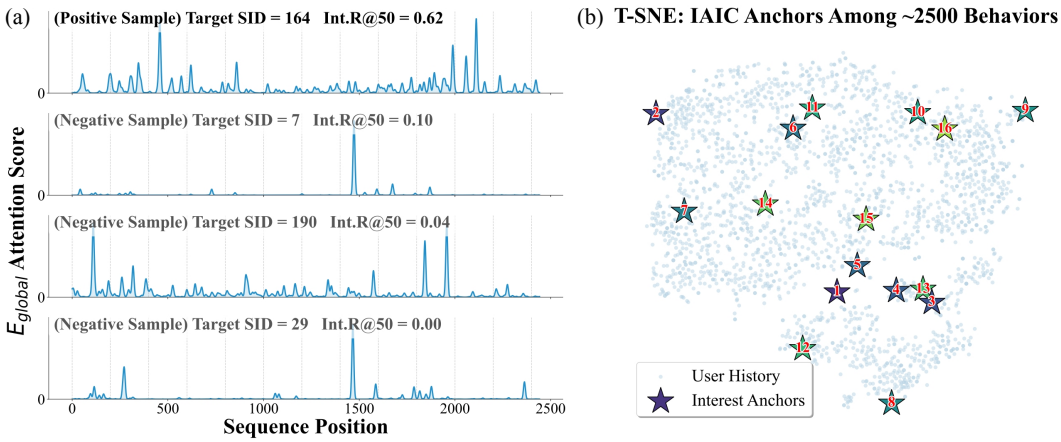


Figure 5: Efficacy of UxSID in interest modeling. (a) highlights the target SID-based attention routing, while (b) showcases the diversity of learned long-term interest anchors.

granularity traditional compression methods struggle to achieve. Figure 5(b) further reveals that the IAIC block captures multiple interest anchors, covering the full spectrum of behaviors. This synergy ensures both the efficiency and comprehensiveness of UxSID in ULSM.

4 Conclusion

In this paper, we propose UxSID, an end-to-end framework that leverages target SIDs for compressed ultra-long sequence modeling in modern recommender systems. UxSID introduces the Item-Agnostic Interest Compression mechanism and Hierarchical Semantic Probing, which operationalize SIDs as semantic-specific routers to navigate complex, ultra-long behavioral histories. By harnessing the high-density semantic resolution of SIDs, the framework effectively bridges the gap between historical interaction density and target intent precision, achieving superior predictive accuracy while maintaining industrial-level efficiency. Extensive offline and online experiments demonstrate that UxSID significantly outperforms SOTA baselines, while comprehensive ablation studies validate the necessity and effectiveness of each constituent module. Future work involves the full-scale deployment in Kuaishou's short-video scenarios and its extension to lifelong sequential modeling.

Acknowledgments and Disclosure of Funding

This work was fully supported by Kuaishou Technology.

References

[1] Rui Zhou, Qinglin Jia, Bo Chen, Peng Xu, Yijia Sun, Siyuan Lou, Chaoxin Fu, Mengyuan Fu, Guoming Shen, Zheli Zhou, et al. A survey of user lifelong behavior modeling: Perspectives on efficiency and effectiveness. 2026.

[2] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, page 2671–2679. ACM, 2019. doi: 10.1145/3292500.3330666.

[3] Rui Zhou, Hao Wang, Wei Guo, Qinglin Jia, Wenjia Xie, Xiang Xu, Yong Liu, Defu Lian, and Enhong Chen. Mit: A multi-tower information transfer framework based on hierarchical task relationship modeling. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 651–660, 2025.

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

[5] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018.

[6] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5941–5948, 2019.

[7] Xue Xia, Pong Eksombatchai, Nikil Pancha, Dhruvil Deven Badani, Po-Wei Wang, Neng Gu, Saurabh Vishwas Joshi, Nazanin Farahpour, Zhiyuan Zhang, and Andrew Zhai. Transact: Transformer-based realtime user action model for recommendation at pinterest. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5249–5259, 2023.

[8] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 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*, pages 2685–2692, 2020.

[9] Jianxin Chang, Chenbin Zhang, Zhiyi Fu, Xiaoxue Zang, Lin Guan, Jing Lu, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, et al. Twin: Two-stage interest network for lifelong user behavior modeling in ctr prediction at kuaishou. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3785–3794, 2023.

[10] Bei Yang, Ke Liu, Xiaoxiao Xu, Renjun Xu, Hong Liu, et al. Learning universal user representations via self-supervised lifelong behaviors modeling. 2021.

[11] Xingmei Wang, Shiyao Wang, Wuchao Li, Jiaxin Deng, Song Lu, Defu Lian, and Guorui Zhou. Transformers are good clusterers for lifelong user behavior sequence modeling. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3123–3132, 2025.

[12] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3702–3712, 2022.

[13] Kirti Jain and Rajni Jindal. Sampling and noise filtering methods for recommender systems: A literature review. *Engineering Applications of Artificial Intelligence*, 122:106129, 2023.

[14] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

[15] Kun Zhang, Jingming Zhang, Wei Cheng, Yansong Cheng, Jiaqi Zhang, Hao Lu, Xu Zhang, Haixiang Gan, Jiangxia Cao, Tenglong Wang, et al. Onemall: One model, more scenarios–end-to-end generative recommender family at kuaishou e-commerce. *arXiv preprint arXiv:2601.21770*, 2026.

[16] Ruining He, Lukasz Heldt, Lichan Hong, Raghunandan Keshavan, Shifan Mao, Nikhil Mehta, Zhengyang Su, Alicia Tsai, Yueqi Wang, Shao-Chuan Wang, et al. Plum: Adapting pre-trained language models for industrial-scale generative recommendations. *arXiv preprint arXiv:2510.07784*, 2025.

[17] Wencai Ye, Mingjie Sun, Shaoyun Shi, Peng Wang, Wenjin Wu, and Peng Jiang. Das: Dual-aligned semantic ids empowered industrial recommender system. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6217–6224, 2025.

[18] Huanjie Wang, Xinchen Luo, Honghui Bao, Zhang Zixing, Lejian Ren, Yunfan Wu, Hongwei Zhang, Liwei Guan, and Guang Chen. Pit: A dynamic personalized item tokenizer for end-to-end generative recommendation. *arXiv preprint arXiv:2602.08530*, 2026.

[19] Junwei Yin, Senjie Kou, Changhao Li, Shuli Wang, Xue Wei, Yinqiu Huang, Yinhua Zhu, Haitao Wang, and Xingxing Wang. Dos: Dual-flow orthogonal semantic ids for recommendation in meituan. *arXiv preprint arXiv:2602.04460*, 2026.

[20] Tian Xia, Jiaqi Zhang, Yueyang Liu, Hongjian Dou, Tingya Yin, Jiangxia Cao, Xulei Liang, Tianlu Xie, Lihao Liu, Xiang Chen, et al. Qarm v2: Quantitative alignment multi-modal recommendation for reasoning user sequence modeling. *arXiv preprint arXiv:2602.08559*, 2026.

[21] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

[22] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 565–574, 2019.

[23] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 540–550, 2022. doi: 10.1145/3511808.3557220. URL `https://doi.org/10.1145/3511808.3557220`.

[24] Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. End-to-end user behavior retrieval in click-through rateprediction model. *arXiv preprint arXiv:2108.04468*, 2021.

[25] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. Sampling is all you need on modeling long-term user behaviors for ctr prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2974–2983, 2022.

[26] Xiang Xu, Hao Wang, Wei Guo, Luankang Zhang, Wanshan Yang, Runlong Yu, Yong Liu, Defu Lian, and Enhong Chen. Multi-granularity interest retrieval and refinement network for long-term user behavior modeling in ctr prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2745–2755, 2025.

[27] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[28] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2400–2409, 2024.

[29] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482*, 2019.

[30] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. User-aware multi-interest learning for candidate matching in recommenders. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1326–1335, 2022.

[31] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2615–2623, 2019.

[32] Chuan He, Yongchao Liu, Qiang Li, Weiqiang Wang, Xing Fu, Xinyi Fu, Chuntao Hong, and Xinwei Yao. Multi-grained preference enhanced transformer for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 872–883, 2025.

[33] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data*, pages 1–4, 2019.

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

[35] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*, 2024.

[36] Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, et al. Mtgr: Industrial-scale generative recommendation framework in meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5731–5738, 2025.

[37] Li-Wei Pan, Wei-Ke Pan, Mei-Yan Wei, Hong-Zhi Yin, and Zhong Ming. A survey on sequential recommendation. *Frontiers of Computer Science*, 20(3):2003606, 2026.

[38] Zihua Si, Lin Guan, ZhongXiang Sun, Xiaoxue Zang, Jing Lu, Yiqun Hui, Xingchao Cao, Zeyu Yang, Yichen Zheng, Dewei Leng, et al. Twin v2: Scaling ultra-long user behavior sequence modeling for enhanced ctr prediction at kuaishou. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4890–4897, 2024.

[39] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 565–574, 2019.

[40] Xin Song, Xiaochen Li, Jinxin Hu, Hong Wen, Zulong Chen, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. Lrea: Low-rank efficient attention on modeling long-term user behaviors for ctr prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2843–2847, 2025.

[41] Wenhan Lyu, Devashish Tyagi, Yihang Yang, Ziwei Li, Ajay Somani, Karthikeyan Shanmugasundaram, Nikola Andrejevic, Ferdi Adeputra, Curtis Zeng, Arun K Singh, et al. Dv365: Extremely long user history modeling at instagram. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4717–4727, 2025.

[42] Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 247–256, 2025.

[43] Xiao Lv, Jiangxia Cao, Shijie Guan, Xiaoyou Zhou, Zhiguang Qi, Yaqiang Zang, Ben Wang, and Guorui Zhou. Marm: Unlocking the recommendation cache scaling-law through memory augmentation and scalable complexity. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 2022–2031, 2025.

[44] Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5915–5922, 2025.

[45] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

[46] Zhen Zhao, Tong Zhang, Jie Xu, Qingliang Cai, Qile Zhang, Leyuan Yang, Daorui Xiao, and Xiaojia Chang. Farewell to item ids: Unlocking the scaling potential of large ranking models via semantic tokens. *arXiv preprint arXiv:2601.22694*, 2026.

[47] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, et al. Onerec-v2 technical report. *arXiv preprint arXiv:2508.20900*, 2025.

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

A Related Work

Foundations of Sequence Modeling and the Scalability Bottleneck. Historically, industrial recommendation systems have adhered to a paradigm that combines sequence modeling with feature interaction. A defining milestone in this trajectory is DIN [5], which introduced a target-aware attention mechanism to dynamically activate historical behaviors relevant to the candidate item. Subsequent innovations expanded on this foundation to capture temporal dependencies (e.g., DIEN [6] and DSIN [29]), mitigate noise within sequences, and extract multi-faceted user interests [30–33]. With the success of Transformer [4], self-attentive architectures such as SASRec [34] have become standard practice. More recently, generative models optimized via next-token prediction, such as HSTU [35] and MTGR [36], have emerged, demonstrating formidable capabilities in learning rich, contextualized user interest representations. However, the quadratic computational complexity inherent in these attentive and generative architectures imposes a severe sequence-length scalability challenge, strictly bounding their deployable length within the stringent millisecond-level latency constraints of online serving environments [1]. This bottleneck inherently truncates the model's receptive field, rendering them inadequate for capturing long-term, heterogeneous user trajectories, thereby necessitating new works specifically tailored for ULSM [37].

Search-Based Paradigms for Ultra-Long Sequence Modeling. To overcome the scalability bottleneck of ultra-long sequences, the industry gravitates toward a two-stage cascaded framework—conceptualized as the "Global Search First, Exact Search Later" paradigm—wherein a lightweight GSU filters the expansive history into a target-relevant top-K sub-trajectory for subsequent fine-grained attention modeling by a more expressive Exact Search Unit (ESU). Early approaches, such as SIM [8], relied on hard filtering based on categorical item attributes, whereas subsequent models introduced hashing and low-precision attention (e.g., ETA [24] and SDIM [25]) to accelerate similarity computation. More recently, TWIN [9] proposed an end-to-end architecture to align the optimization objectives of both stages, a framework that was subsequently enhanced in TWINv2 [38] via hierarchical clustering to further boost scalability and performance.

Despite their success, search-based methods suffer from an inherent limitation: their filtering logic often sacrifices historical items with low superficial correlation but high latent semantic synergy [26]. For example, a search unit biased toward "pants" may discard a matching belt or stylistic footwear, missing crucial combinatorial intents. In contrast, SIDs transcend these surface-level filters, capturing high-order semantic links often invisible to traditional retrieval [14, 15].

Compression-Based Paradigms for Ultra-Long Sequence Modeling. To circumvent the prohibitive online computational costs of search-based methods and better exploit the global contextual information inherent in long sequences, compression-based approaches [2, 10, 12, 39–43] have emerged. These methods primarily aim to offload the heavy burden of sequence modeling to the offline stage, compressing ultra-long sequences into compact, reusable user representations. Early explorations include memory-augmented models like MIMN [2] and HPMN [39], which leverage memory networks to effectively compress sequences at the scale of thousands of interactions offline. Recent advancements focus on modularizing and scaling this offline representation learning. For instance, LREA [40] employs low-rank matrix factorization to compress sequence representations, enabling downstream models to efficiently capture user interests via target attention over these condensed embeddings. Alternatively, rule-based partitioning methods, such as DV365 [41], slice sequences into fixed-length or temporal windows. PinnerFormer [12], LURM [10], and C-Former [11] condense expansive user sequences into interest clusters or dense embeddings.

While compression-based methods excel in online efficiency, static compression inherently acts as a low-pass filter—retaining coarse-grained global trends while marginalizing the high-frequency, target-specific interest peaks critical for precise ranking. This limitation underscores a pressing need for adaptive compression frameworks capable of generating target semantic-aware representations.

Item Tokenization and Semantic Identifiers. Item identifiers are fundamental to modern recommender systems. Current paradigms are transitioning from random Photo IDs (PIDs) [5, 8], which rely on embedding lookups, toward SIDs constructed via RQ-VAE [14] or other tokenization frameworks (e.g., QARM [44]). Distinct from existing paradigms that primarily utilize SIDs for generative recommendation [45, 16, 18] or item representation enrichment [20, 46], we propose to employ SIDs as dynamic semantic queries within the ULSM context. Their clustered semantic codes enable navigating complex interest spaces where traditional PID-based models lack sufficient granularity.

B Online Implementation

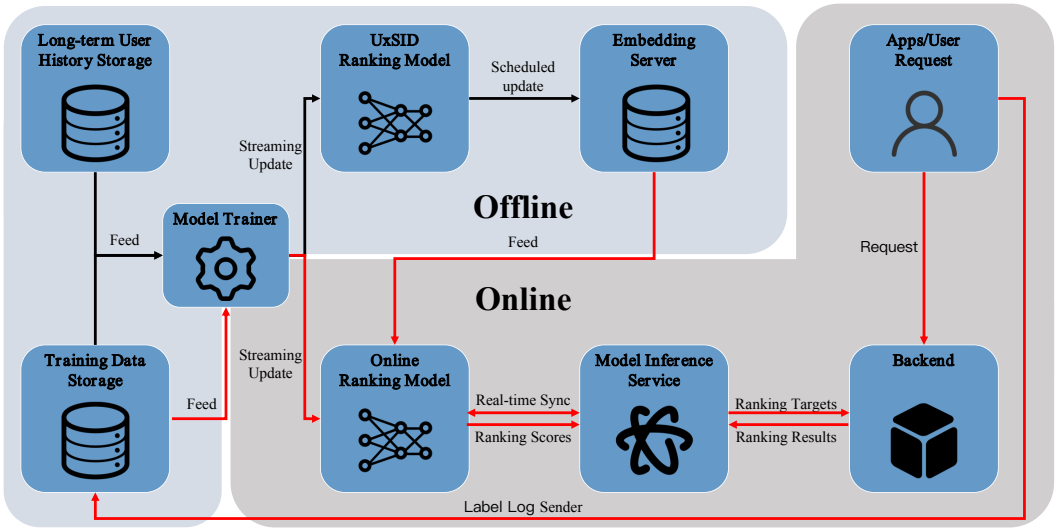


Figure 6: The overall system deployment pipeline of UxSID, comprising offline UxSID embedding generation (black path), online model training (red path), and real-time online inference (red path).

Model Configurations and Maintenance. UxSID utilizes the same SIDs codebook architecture as QARM V2 [20] and OneRec-v2 [47] , which has demonstrated outstanding performance across multiple industrial scenarios. Specifically, the first-level codebook size is set to 4096, with an embedding lookup dimension of 32. This dimension is strategically aligned with the hidden units of the three single-head attention networks within UxSID to ensure representation consistency. To maintain the timeliness of the semantic representations, the embedding server undergoes a scheduled update once per week. The comprehensive end-to-end pipeline, encompassing both offline generation and online serving, is illustrated in Figure 6.

Computational Cost and Latency Analysis. We evaluate the resource efficiency of UxSID through both training and inference stages. For the UxSID model offline training, a sequence length of 1k requires 16 GPUs, while scaling to a 10k sequence length necessitates 40 GPUs. Crucially, the resource consumption of the downstream online model remains invariant to the sequence length, as it only interacts with the compressed UxSID embeddings (of shape $[2, 32]$) in an incremental manner. In terms of serving infrastructure, while the current online model with a 1k sequence length occupies approximately 450 GPUs, we estimate that directly scaling the sequence length to 10k within the ranking model would necessitate approximately 5,300 GPUs to meet throughput requirements. By shifting the computational burden of long-sequence modeling to the UxSID framework, our approach introduces a negligible latency overhead of only +0.16 ms compared to the baseline without UxSID embeddings. This balance between high-capacity modeling and minimal inference latency underscores the efficiency of UxSID for real-time recommendation at scale.

C Datasets

Table 6: Statistics of public datasets.

Dataset	#Users	#Items	#Interaction	Avg. Seq. Length	Max. Seq. Length
XLong	1,000	3,269,017	1,000,000	1,000	1,000
KuaiRec-Big	7,176	10,728	12,530,806	1,746	3,000

XLong [2] **[22]:** Sampled from a large e-commerce platform (April–September 2018), this dataset provides sequences of about 1k interactions per user for long-term modeling. For each item sequence, the last video is used for testing, the video before the last is used for validation, and the rest is used for training.
KuaiRec-Big [3] **[23]:** KuaiRec is a real-world dataset collected from the recommendation logs of the app Kuaishou, containing user interactions and video metadata from July 5, 2020 to September 5, 2020. We use users' interaction histories to create video sequences sorted by timestamp and filter out users with less than 50 comments. For each video sequence, the last 7 videos are used for testing, the 8th to 14th last videos are used for validation, and the rest are used for training. We fix the user's interaction history up to 2k.
Industrial Dataset: Collected from Kuaishou's advertising system (April 1-7 2026), this dataset contains impression logs and labels. For each user, the interaction history is preserved up to 10k.
SIDs Generation. For the KuaiRec-Big dataset, we extract item-related fields from item_daily_features.csv and employ the Llama model [48] to generate high-dimensional content embeddings. The XLong dataset comes with pre-provided item embeddings, which we utilize directly. For both datasets, we apply the quantization toolkit provided by LETTER [4][28] to train the hierarchical codebooks, resulting in SIDs with a structured configuration of $[256 \times 256 \times 256 \times 256]$.

D Baselines

Below we briefly describe baselines with which we compare UxSID:

- **DIN [5]** is a foundational baseline that introduces a target-aware attention mechanism from recent behavior sequences.
- **SIM Hard / Soft** [8] adopts a two-stage approach where a GSU filters the expansive history via either category-based hard matching or embedding-based soft retrieval.
- **ETA** [24] employs Locality Sensitive Hashing and Hamming distance for GSU retrieval.
- **SDIM** [25] adopts a sampling-based approach that utilizes multiple hash functions to generate signatures for ULSM.
- **MIRRN** [26] utilizes multi-granularity queries across diverse time scales for behavior retrieval.
- **TWIN** [9] ensures target consistency between the GSU and ESU stages for ULSM.
- **C-Former** [11] employs reconstruction constraints to cluster user interests in an end-to-end manner.

E Detailed Results on Public Datasets

E.1 Robustness and Stability Analysis

To evaluate the stability of UxSID, we report the performance across three independent runs using different random seeds (2024, 2025, and 2026). As shown in Table 7, the marginal variance observed across these runs demonstrates that our framework is highly robust against initialization noise and consistently maintains its performance gains across different experimental trials.

Table 7: Detailed performance of UxSID across three independent runs with different random seeds (2024, 2025, 2026). The final results reflect the stability of our proposed framework.

Model	Seed	KuaiRec-Big (AUC)	XLong (AUC)
UxSID	2024	0.83484	0.84084
UxSID	2025	0.83482	0.84405
UxSID	2026	0.83462	0.83964
Avg $\pm$ Std	-	**0.8348 $\pm$ 0.0001**	**0.8415 $\pm$ 0.0023**

[2] https://tianchi.aliyun.com/dataset/22482
[3] https://kuairec.com/
[4] https://github.com/HonghuiBao2000/LETTER/tree/master

E.2 Performance Gain Attribution

A potential concern is whether the performance gain of UxSID stems primarily from the item-side SID information rather than the proposed architecture. To isolate the impact of this information enrichment, we conduct a controlled experiment where the first-layer SID is incorporated as an additional sparse feature for all baseline models.

Table 8 shows that while the inclusion of SID features slightly improves the performance of all baselines, UxSID consistently maintains a substantial lead. This confirms that our superiority is fundamentally derived from the structural design of the IAIC and the hierarchical routing mechanism, rather than mere information enrichment.

Table 8: Comparison with baselines after augmenting all models with item-side SID features. The best results are in bold.

Models (+ SID Features)	**XLong**	**KuaiRec-Big**
DIN	0.7932	0.8214
SIM-Hard	–	0.8246
SIM-Soft	0.7999	0.8305
ETA	0.7930	0.8271
SDIM	0.7955	0.8246
MIRRN	0.7976	0.8257
TWIN	0.8189	0.8296
C-Former	0.8180	0.8311
UxSID (Base)	0.8408	0.8348
UxSID + SID	**0.8439**	**0.8361**

F Limitation

Despite its efficiency, the industrial deployment of UxSID presents certain specialized requirements that could be challenging in general settings. First, the framework's performance is intrinsically linked to the quality of pretrained SIDs; in massive-scale production environments, ensuring the real-time availability of SIDs for cold-start items (i.e., fetching SIDs during the very first user request) requires a highly optimized streaming inference pipeline. Second, maintaining the large-scale storage and high-concurrency retrieval efficiency of SID embeddings poses non-trivial engineering hurdles. These remain key considerations for adapting UxSID to other large-scale recommendation systems.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline our contributions, including the UxSID framework for ultra-long sequence modeling and its constant-time inference capability recommendation systems.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Appendix F, such as the reliance on high-quality semantic embeddings for SID generation and potential storage challenges during large-scale deployment.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper focuses on empirical framework design and complexity analysis rather than abstract theoretical theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of the model architecture (Section 2), hyperparameters (Section 3.5), and the data preprocessing pipeline for public datasets in the Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Due to company policies, we have not yet made our source code publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Full details on data splits, optimization settings (e.g., Adam optimizer, learning rates), and hardware configurations are specified in Section 3. The data preprocessing pipeline for public datasets in the Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our main results on public benchmarks are reported as the average of multiple runs with standard deviations (as shown in Table 1 and Table 7).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For industrial deployment, we specify the use of GPUs for both training and inference experiments. We provide comprehensive details regarding memory utilization and the incremental overhead in online inference latency.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work follows standard practices for recommendation system research and does not violate any ethical guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: This work focuses on the fundamental architectural optimization of ultra-long sequence modeling. As a core algorithmic framework, its societal impact is indirectly determined by the specific downstream applications it is integrated into, making a generalized discussion of societal impacts premature and beyond the scope of this technical contribution.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our model is a structural framework for recommendation and does not pose high risks for misuse like large generative models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite and credit the creators of the XLong and KuaiRec-Big datasets and follow their respective terms of use (Section 3). All baseline models are cited and

implemented following their respective original publications with default settings to ensure a fair and reproducible comparison (Appendix C).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The details of our proposed UxSID framework are thoroughly documented in the main text and the appendix of this paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This research does not involve crowdsourcing or human subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research uses anonymized historical interaction logs and does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We disclose in Appendix C that Llama-7b was used as a feature extractor to generate content embeddings for items in public datasets.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.