Article

Not peer-reviewed version

A Survey of User Lifelong Behavior Modeling: Perspectives on Efficiency and Effectiveness

Rui Zhou †, Qinglin Jia †, Bo Chen, Peng Xu, Yijia Sun, Siyuan Lou, Chaoxin Fu, Mengyuan Fu, Guoming Shen, Zheli Zhou, Jinlong Jiao, Naifu Zhou, Shijie Guan, Yunjing Qi, Shiyao Wang, Xinchen Luo, Qigen Hu, Chaoyi Ma, Xiao Lv, Qiang Luo, Yuyang Ye, Luankang Zhang, Defu Lian, Ruiming Tang *, Guorui Zhou, Han Li, Kun Gai, Hao Wang *, Enhong Chen *

Posted Date: 21 January 2026

doi: 10.20944/preprints202601.1559.v1

Keywords: user lifelong behavior modeling; recommender systems; heterogeneous behavior sequences; efficiency–effectiveness trade-off; industrial-scale recommendation

Article

A Survey of User Lifelong Behavior Modeling: Perspectives on Efficiency and Effectiveness

Rui Zhou [1,†], Qinglin Jia [2,†], Bo Chen[2], Peng Xu [2], Yijia Sun [2], Siyuan Lou [2], Chaoxin Fu [2], Mengyuan Fu [2], Guoming Shen [2], Zheli Zhou [2], Jinlong Jiao [2], Naifu Zhou [2], Shijie Guan [2], Yunjing Qi [2], Shiyao Wang [2], Xinchen Luo [2], Qigen Hu [2], Chaoyi Ma [2], Xiao Lv [2], Qiang Luo [2], Yuyang Ye [1], Luankang Zhang [1], Defu Lian[1], Ruiming Tang [2,*], Guorui Zhou[2], Han Li[2], Kun Gai[3], Hao Wang[1,*] and Enhong Chen[1,*]

[1] University of Science and Technology of China
[2] Kuaishou Technology
[3] Independent Researcher
* Correspondence: tangruiming@kuaishou.com; wanghao3@ustc.edu.cn; cheneh@ustc.edu.cn
† Rui Zhou and Qinglin Jia contribute equally to this work.

Abstract

In industrial-scale recommender systems, the continuous accumulation of user interactions gives rise to large-scale and heterogeneous behavior sequences, posing significant challenges to both computational efficiency and storage scalability. To support user lifelong behavior modeling (ULBM) under stringent industrial constraints, extensive research efforts have been devoted to balancing efficiency and effectiveness. This survey presents a systematic review of ULBM methods that have been widely adopted in real-world recommender systems and demonstrated substantial practical value. We organize existing studies around the central industrial challenge of efficiency–effectiveness trade-offs. Specifically, efficiency is examined from both algorithmic and system-level perspectives, while effectiveness is discussed in terms of enhanced modeling of intrinsic sequential dependencies and the incorporation of external contextual signals. We further highlight how the synergy between efficiency-oriented and effectiveness-oriented designs continually improves the return on investment in large-scale recommender systems. Finally, we summarize publicly available datasets for ULBM research and outline several promising directions for future investigation, aiming to provide insights and guidance for subsequent studies. To support ongoing research, we maintain a living repository tracks emerging literature and reference implementations: https://github.com/Kuaishou-RecModel/Survey_of_ULBM.

Keywords: user lifelong behavior modeling; recommender systems; heterogeneous behavior sequences; efficiency–effectiveness trade-off; industrial-scale recommendation

1. Introduction

Systems (RS) are a vital bridge between the vast amount of information on the Internet and its users, facilitating billions of daily interactions across various Internet platforms such as e-commerce (e.g., Amazon, Taobao) [1,2], video sharing (e.g., YouTube, TikTok, Kuaishou) [3,4], and social media (e.g., Facebook, Instagram) [5,6]. By leveraging users' historical behavioral signals (e.g., clicks, likes, follows), RS estimate the likelihood of future user–item engagements and generate ranked item lists that constitute the final recommendations [7–9].

In real-world applications, recommender systems often operate over billions of candidate items [10–13], rendering exhaustive scoring of all user–item pairs computationally impractical. As a result, a cascaded retrieval and ranking paradigm has become the standard solution [14–16]. In this paradigm, retrieval refers to the use of lightweight models to efficiently narrow the search space by selecting a small set of promising items from an extremely large corpus, typically on the order of

 doi:10.20944/preprints202601.1559.v1

thousands [17–19]. Ranking then applies more expressive models to these retrieved candidates for fine-grained scoring and ordering [20–22], enabling the paradigm to balance efficiency and effectiveness in large-scale recommendation.

In this cascaded paradigm, user historical behaviors constitute the central signal for modeling user interests and play a decisive role in determining the effectiveness of both retrieval and ranking. With the continued growth of Internet services, user behavior logs are accumulating at an unprecedented scale. On large-scale video-sharing platforms such as Kuaishou, users may interact with hundreds of videos per day, leading to lifelong behavior sequences with lengths reaching the order of 10^5 [23]. Consequently, **User Lifelong Behavior Modeling** (ULBM) has become a prominent research focus in industrial-scale recommender systems [24–26]. Here, we define ULBM as the problem of modeling ultra-long and heterogeneous user behavior sequences under strict industrial efficiency constraints. However, applying ULBM in real-world recommender systems is challenging, as more expressive behavior models often incur substantial computational and serving overheads. In this setting, efficiency under stringent industrial constraints and effectiveness in improving recommendation accuracy emerge as two tightly coupled yet conflicting objectives. This fundamental tension gives rise to the concept of the **Efficiency–Effectiveness Balance** (EEB), which we consider a central principle in ULBM research.

In this context, user lifelong behavior sequences exhibit two salient characteristics: (I) ***Ultra-long sequences***. User behavior sequences in industrial recommender systems have expanded to tens of thousands of interactions for a large proportion of users, who collectively account for the majority of platform engagements. As shown in Figure 1(a), one-third of users with more than 10,000 historical interactions contribute over 95% of the total watch time. Such ultra-long sequences fundamentally violate the assumptions underlying most short-sequence modeling methods, which assume that user preferences can be adequately captured by a fixed-length, temporally localized interaction context [27,28]. As sequence length grows from thousands to hundreds of thousands, both computational complexity and memory consumption increase dramatically, rendering naive sequence encoding infeasible in large-scale systems. (II) ***Heterogeneous behaviors***. User interactions are inherently heterogeneous, encompassing multiple behavior types such as clicks, likes and shares. As illustrated in Figure 1(b), the majority of users exhibit more than five types of active behaviors. These heterogeneous signals differ substantially in intent strength, temporal dynamics, and semantic meaning, thereby violating the homogeneous interaction assumption underlying many traditional sequence modeling approaches. As a result, naive sequence concatenation or uniform attention weighting becomes inadequate, since it fails to distinguish varying signal strengths and cross-behavior semantics. Effectively integrating such diverse behaviors thus requires fine-grained modeling mechanisms that can capture heterogeneous contributions and complex cross-behavior dependencies, inevitably increasing modeling complexity [29,30].

Based on the above discussion, the combination of ultra-long sequences and heterogeneous behaviors makes user lifelong behavior histories an exceptionally rich and complex source of information for modeling user interests. As illustrated in Figure 1(c), the continual accumulation of diverse behavioral signals introduces increasingly informative contextual cues, enabling sustained performance improvements when effectively modeled. This observation highlights the critical role of ULBM in industrial-scale recommender systems, where advances in modeling capacity must continuously balance information exploitation against strict efficiency constraints.

 doi:10.20944/preprints202601.1559.v1

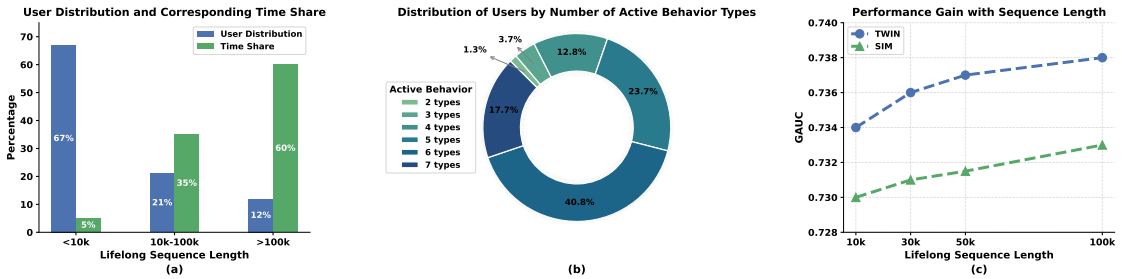

Figure 1. (**a**) User distribution based on lifelong sequence length and their share of total viewing duration. Highly active users (>100k interactions) account for the largest share of total viewing duration [31]. (**b**) Distribution of users by the number of active behavior types within one month, including six positive and one negative feedback types. Most users(82.2%) exhibit five or more behavior types [32]. (**c**) Model performance with increasing lifelong behavioral sequence length. Both **SIM** and **TWIN** serve as representative baselines, showing consistent performance gains as sequence length increases [33].

1.1. Efficiency-Effectiveness Balance in RS

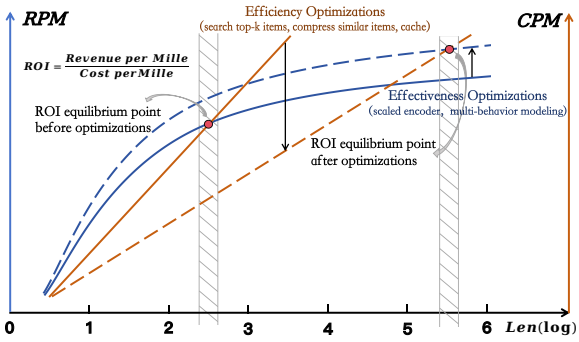

Figure 2. Efficiency optimization strategies reduce CPM, while effectiveness optimization strategies increase RPM. Their combined impact shifts the ROI frontier upward and outward as sequence length increases.

Industrial-scale recommender systems operate under stringent efficiency constraints, typically requiring end-to-end latency within hundreds of milliseconds while sustaining hundreds of thousands of queries per second [24,34]. Such constraints substantially limit the applicability of computationally expensive models, even when they offer potential gains in recommendation effectiveness. As a result, modeling user lifelong behavior sequences must contend with an inherent tension between efficiency and effectiveness, which we formalize as the Efficiency–Effectiveness Balance (EEB).

EEB characterizes a constrained optimization landscape in industrial-scale recommender systems, where improvements in advancing interaction modeling capability, fine-grained user interest understanding, and incorporating worldwide knowledge must be achieved under strict latency, memory, and throughput constraints. The pursuit of EEB has driven extensive research into methods [35–37] that jointly optimize efficiency and effectiveness, which is essential for maintaining high return on investment (ROI) in real-world deployments.

Figure 2 illustrates how the ROI equilibrium point varies with Revenue per Mille (RPM) and Cost per Mille (CPM). As user behavior sequences grow longer, our goal is for the optimized ROI equilibrium point to shift to the right. ROI, defined as the ratio of RPM to CPM, can be improved either by reducing CPM or by increasing RPM. Efficiency-oriented optimization primarily targets CPM, producing a roughly linear decrease in cost through sequence summarization, algorithmic, and infrastructural strategies, while effectiveness-oriented optimization focuses on RPM, often resulting in a logarithmic growth by leveraging richer user histories and contextual or external signals. These two dimensions constitute the core methodological axes of ULBM, which we review in Sec. 3 for efficiency-oriented techniques and Sec. 4 for effectiveness-oriented advances.

 doi:10.20944/preprints202601.1559.v1

1.2. Contribution of This Survey

ULBM in industrial-scale recommender systems has made substantial progress in recent years, with growing efforts devoted to balancing efficiency and effectiveness under real-world constraints. However, the resulting literature has become increasingly extensive and heterogeneous, calling for a systematic and unified understanding from an industrial perspective.

Table 1 provides a structured overview of existing surveys on user behavior modeling, organizing them along four key scopes that are critical for industrial recommender systems, namely industrial deployment, ultra-long sequences, heterogeneous behaviors, and sequence modeling. Although these surveys offer valuable insights, their coverage remains fragmented when viewed through the lens of industrial-scale deployment. Early works around 2023, such as He et al. [25] and Chen et al. [38], primarily focused on heterogeneous behavior and sequence modeling, with limited attention to ultra-long sequences and deployment considerations. Subsequent surveys gradually expanded their scope toward longer behavior sequences and richer modeling paradigms. Pan et al. [39] emphasized ultra-long sequence modeling, while Chen et al. [40] focused on cross-domain sequential recommendation. More recent work by Kim et al. [41] further systematized frameworks and pipelines for multi-behavior sequential recommendation. Despite this progression, existing surveys still lack a truly unified perspective that jointly addresses ultra-long sequences, heterogeneous behaviors, and practical industrial deployment constraints.

In contrast to existing surveys, our work focuses on user lifelong behavior modeling under industrial deployment, with ultra-long temporal spans and heterogeneous behavioral signals. Beyond coverage, our survey departs from prior work in its organizing principle. Rather than categorizing methods by model architectures or learning paradigms, we reorganize the literature from the perspective of EEB, which fundamentally constrains industrial-scale recommender systems.

We conducted a comprehensive literature review covering major venues in recommender systems and data mining. To ensure both methodological representativeness and practical relevance, we place particular emphasis on studies developed and evaluated in industrial-scale recommender systems, especially those validated on real-world production data or demonstrated through online deployment and large-scale A/B testing. This focus allows the survey to reflect not only algorithmic advances but also the practical constraints and design considerations faced in real-world deployments. While motivated by industrial-scale recommender systems, the challenges and solutions discussed in this survey reveal fundamental issues in sequence modeling and representation learning.

In summary, our main contributions are as follows:

Table 1. Comparison of Existing Surveys.

Survey \ Scope	Industrial Deployment	Ultra-long Sequences	Heterogeneous Behaviors	Sequence Modeling
He et al.,2023 [25]	○	○	●	●
Chen et al.,2023 [38]	○	○	●	●
Pan et al.,2024 [39]	○	●	○	●
Chen et al.,2024 [40]	○	○	○	●
Kim et al.,2025 [41]	○	○	●	●
Ours	●	●	●	●

- We provide the first unified and industrial-oriented survey on User Lifelong Behavior Modeling (ULBM) in large-scale recommender systems. From a lifecycle-aware perspective, we systematically review how ultra-long and heterogeneous user behavior sequences are modeled under real-world efficiency constraints, bridging the gap between academic formulations and industrial deployment.
- We formalize the Efficiency–Effectiveness Balance (EEB) as a central analytical framework for characterizing and comparing ULBM methods. By viewing existing approaches through the lens of constrained optimization, we reveal how different modeling paradigms embody distinct trade-

 doi:10.20944/preprints202601.1559.v1

offs between computational efficiency and representation expressiveness, offering a principled basis for method comparison.
- We distill common design patterns, implicit assumptions, and inherent limitations across industrial ULBM approaches, and summarize practical deployment insights from real-world systems. Based on these observations, we identify open challenges and promising research directions to guide future advances in lifelong user modeling.

1.3. Survey Organization

The remainder of this paper is organized as follows. Section 2 introduces the preliminary concepts and research background. Section 3 examines efficiency optimization, including both algorithmic and infrastructure-level techniques. Section 4 discusses effectiveness optimization, including interaction modeling, fine-grained modeling, and external knowledge incorporation. Section 5 summarizes the commonly used datasets in user lifelong behavior modeling, while Section 7 highlights potential directions for future research in this field. Finally, Section 8 concludes the paper.

2. Background and Preliminaries

To help readers better understand ULBM and its deployment in industrial-scale recommender systems, and to lay the groundwork for subsequent discussions, we first provide the problem formulation of ULBM. We then present an overview of a simplified system architecture, representative of typical industrial recommender system deployments. Finally, we review commonly used techniques in behavior sequence modeling, providing the necessary background knowledge and preliminary concepts.

2.1. Problem Formulation of ULBM

User lifelong behavior modeling (ULBM) aims to capture the continuously accumulating interactions of a user over time, forming the foundation for understanding long-term preferences. Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. For a user $u \in \mathcal{U}$, the lifelong behavior sequence is defined as $S = [b_1, b_2, \ldots, b_L]$, where each behavior $b_k = (i_k, a_k, t_k, c_k)$ represents an interaction with item i_k, the type of behavior a_k (e.g., click, like, comment), the timestamp t_k, and associated contextual information c_k. The heterogeneous nature of b_k influences the design of embedding representations, the choice of aggregation operators, and the semantics of attention weights, rendering naive sequence concatenation or uniform weighting inadequate. Furthermore, each interaction is mapped to a learnable embedding $\mathbf{b}_k \in \mathbb{R}^d$, forming the token-level sequence $\mathbf{S} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L]$. Here, $\mathbf{b}_k$ serves as the fundamental modeling unit (token), and $\mathbf{S}$ represents the user's lifelong behavior embedding. For notational simplicity, we omit the user subscript when there is no ambiguity.

Based on the token-level sequence $\mathbf{S}$, ULBM in industrial-scale recommender systems operates under EEB, which trades off computational cost and model expressiveness. In practice, ULBM can be applied at different stages of the recommendation pipeline, but the majority of methods focus on the ranking stage, where $\mathbf{S}$ is used to produce user interest representations for more precise recommendations.

2.2. ULBM in ranking system deployment

ULBM finds broad applications in modern recommender systems, but in industrial deployments it is primarily applied in the ranking stage. A complete ranking pipeline typically consists of multiple modules, including long-term behavior modeling, short-term behavior modeling, feature interaction, and others, with ULBM responsible for capturing ultra-long and heterogeneous user behaviors in order to produce precise user representations.

To illustrate its practical usage, we briefly outline a simplified industrial ranking pipeline, in which ULBM serves as the core component. In industry, the ranking model pipeline for online serving is broadly divided into **offline training** and **online serving**. Offline training, typically conducted on distributed machine learning platforms, focuses on constructing and optimizing recommendation

 doi:10.20944/preprints202601.1559.v1

models to produce high-performance models ready for deployment. Once trained, these models are deployed to high-throughput online serving systems, which must handle massive traffic under strict latency constraints. For instance, at Taobao [42], the system serves over 120,000 queries per second (QPS) with a latency of less than 30 ms, while Kuaishou [43] supports more than 400,000 QPS with latency under 500 ms. Incorporating lifelong user behavior further complicates real-time serving in industrial settings, as long-term user interest modeling introduces significant challenges in storage and latency. The computational overhead increases rapidly with the length of user behavior sequences, creating potential bottlenecks for such models. These challenges occur in both offline training and online serving, prompting a more detailed discussion of each stage in the following subsubsections.

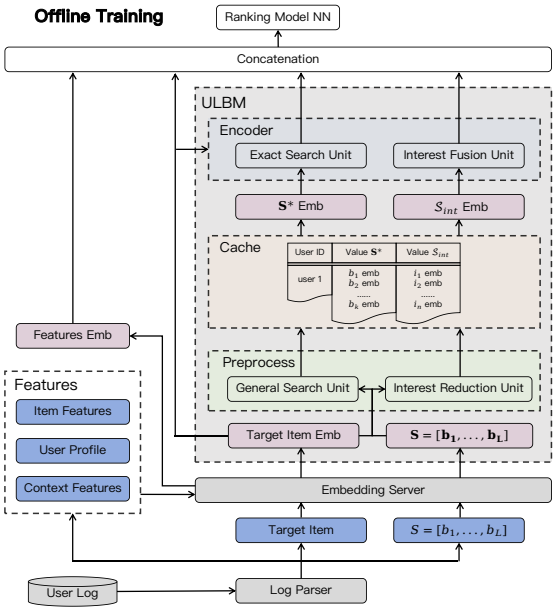

Figure 3. Workflow of ULBM for Offline Training.

2.2.1. Offline Training

The offline training workflow of ULBM constitutes a comprehensive pipeline engineered for systematically processing user behavior data and generating precise ranking predictions. Figure 3 illustrates the offline training workflow. Specifically, the architecture orchestrates four primary components: a data pipeline for training sample generation, an embedding server for feature management, a ULBM module for user interest encoding, and a ranking module for engagement estimation.

From a macro perspective, the system is engineered to tackle the challenge of processing extensive user behavior sequences. Large-scale industrial platforms report that a non-negligible portion of users accumulate thousands of interactions within a few months [42]. To efficiently model long-term user interests under practical inference constraints, ULBM frameworks commonly adopt a hierarchical two-stage paradigm. In the first stage, the effective scale of the original behavior sequence $\mathbf{S}$ is reduced through search-based or compression-based modules, yielding a compact representation. Representative implementations include general search units (GSU) and interest reduction units (IRU), which typically produce either a refined subsequence $\mathbf{S}^*$ containing the top-k behaviors most relevant to the target item, or a compressed user interest set $\mathcal{S}_{int}$. The terms "target item" and "candidate item" are used interchangeably in this context.

In the second stage, these intermediate representations are further processed by an interest encoder (e.g., ESU or IFU), which models fine-grained interactions between the target item and the intermediate representation to generate a compact user interest embedding. This embedding is subsequently concatenated with other feature embeddings (e.g., user profiles, item attributes, and contextual signals), and then fed into the ranking module to produce the final prediction.

Beyond the two-stage ULBM workflow, additional system components are required to support efficient training and overall pipeline operation. The data pipeline constructs training samples by

 doi:10.20944/preprints202601.1559.v1

merging historical and recently collected interaction logs, incorporating item attributes, user profiles, contextual features, and lifelong behavior sequences. The embedding server maintains and updates embeddings for sparse categorical features. During offline training, the structured intermediate representations produced by GSU or IRU (i.e., the embedding subsequence $\mathbf{S}^*$ or the interest embedding set $\mathcal{S}_{int}$) can be materialized in a cache, enabling reuse during online inference without repeated interest search or reduction.

2.2.2. Online Serving

As illustrated in Figure 4, upon receiving a request, the online serving system first fetches the required feature embeddings from the embedding server and retrieves the precomputed sequence embeddings from the cache(we note that this represents one common ULBM paradigm; some methods perform interest search or compression in real-time using the original sequence S instead of relying on a cache). This caching mechanism effectively alleviates computational bottlenecks during online inference. The condensed sequence representation is then encoded into a compact vector, which is subsequently concatenated with other feature embeddings to form the input to the ranking module, which finally outputs the engagement probability.

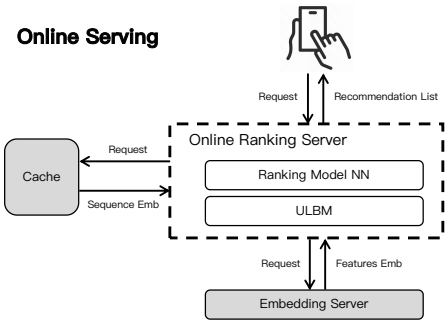

Figure 4. System Architecture for Online Serving.

2.3. Fundamental Techniques for Behavior Modeling

In this subsection, we review the fundamental techniques for modeling user behavior sequences and examine their applicability to ULBM.

Early recommender systems typically modeled user histories of moderate length, where pooling-based [44], attention-based [45,46], and graph-based methods [47,48] were sufficient to capture short-term user preferences. When extended to lifelong behavior sequences with tens of thousands of heterogeneous interactions, however, these methods face fundamental challenges, as their assumptions on homogeneous signals, limited temporal context, and affordable computation no longer hold.

To address these challenges, industrial ULBM systems commonly introduce an explicit preprocessing stage to balance efficiency and effectiveness, with search-based and compression-based methods dominating real-world deployments. These developments motivate a closer examination of the core encoding techniques, such as pooling, attention, and graph-based methods, that underpin modern ULBM frameworks.

2.3.1. Pooling-based Methods

These methods refer to a class of techniques that transform the sequences $\mathbf{S}$ of variable length into the $\mathbf{h} \in \mathbb{R}^d$ through aggregation operations. Typical examples include average pooling, expressed as $\mathbf{h}_{\text{avg}} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{b}_i$; sum pooling, expressed as $\mathbf{h}_{\text{sum}} = \sum_{i=1}^{L}\mathbf{b}_i$; and max pooling, expressed as $\mathbf{h}_{\max}[i] = \max_{j=1,\ldots,L}\mathbf{b}_j[i], \quad i = 1,\ldots,d$, where d is the dimension of the $\mathbf{b}_i$.

Despite their computational efficiency, pooling-based methods inherently assume homogeneous interactions and static importance across the entire sequence. As a result, they fail to capture temporal decay, behavior heterogeneity, and fine-grained intent variations that are critical in lifelong behavior modeling [45,49,50].

 doi:10.20944/preprints202601.1559.v1

2.3.2. Attention-based Methods

These methods refer to a class of techniques that dynamically assign different importance weights to individual token within the sequence $\mathbf{S} \in \mathbb{R}^{L\times d}$ when generating a representation $\mathbf{h} \in \mathbb{R}^d$.

Let $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ denote the learnable query, key, and value matrices, respectively, and let d be the embedding dimension. Softmax is a normalization activation that converts scores into a probability distribution summing to 1. Formally, the attention operation produces an output matrix $\mathbf{H}$ whose shape depends on the query $\mathbf{q}$:

$$\begin{aligned} &\mathbf{Q} = \mathbf{q}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{S}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{S}\mathbf{W}_V, \\ &\mathbf{H} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}. \end{aligned} \tag{1}$$

The primary distinction among different attention mechanisms lies in the form of the query $\mathbf{q} \in \mathbb{R}^{|\mathbf{q}|\times d}$, leading to a unified computational complexity of $O(|\mathbf{q}|Ld)$. In target attention, the query corresponds to the target item embedding, i.e., $|\mathbf{q}| = 1$, and the attention output $\mathbf{H} \in \mathbb{R}^{1\times d}$ can be directly treated as a fixed-dimensional user representation $\mathbf{h} \in \mathbb{R}^d$. In cross attention, the query consists of multiple tokens (e.g., target item and user profiles), whereas in self-attention, the query is the user behavior sequence itself ($\mathbf{q} = \mathbf{S}$). For both cross and self-attention, $\mathbf{H} \in \mathbb{R}^{|\mathbf{q}|\times d}$ undergoes a transformation to produce a fixed-dimensional representation $\mathbf{h} \in \mathbb{R}^d$ for downstream modeling.

Building on the attention mechanisms introduced above, Transformer-based architectures [51] further model user sequences $\mathbf{S}$ by stacking multiple self-attention layers and feed-forward networks. Each layer produces sequence-level representations $\mathbf{H} \in \mathbb{R}^{L\times d}$, which are subsequently aggregated into a fixed-dimensional user embedding $\mathbf{h} \in \mathbb{R}^d$ for downstream tasks. The computational complexity of each Transformer block is $O(L^2 d)$, dominated by self-attention, which poses significant challenges for ultra-long sequences in industrial ULBM [52,53] and motivates the use of preprocessing or approximation techniques to strike a balance between computational efficiency and representation effectiveness.

While attention-based methods offer strong expressiveness and have become a standard choice for short- and medium-length sequence modeling [51,54], their applicability to ULBM is fundamentally constrained by both efficiency and scalability. In particular, self-attention incurs quadratic complexity with respect to sequence length, rendering it infeasible for lifelong behavior sequences without aggressive truncation, sampling, or hierarchical redesign. Moreover, naive attention mechanisms implicitly treat heterogeneous behaviors as comparable tokens, obscuring differences in intent strength and temporal semantics across behavior types.

2.3.3. Graph-based Methods

Graph-based methods represent a user sequence $\mathbf{S} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L]$ as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ corresponds to an interaction token $\mathbf{b}_i$, and edges $(v_i \to v_j) \in \mathcal{E}$ encode temporal or semantic dependencies. Node embeddings are updated via message passing:

$$\mathbf{h}_i^{(r+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\,\mathbf{W}^{(r)}\mathbf{h}_j^{(r)}\right), \quad r = 0, \ldots, R-1, \tag{2}$$

where $\mathbf{h}_i^{(0)} = \mathbf{b}_i$, $\mathcal{N}(i)$ denotes the neighbors of node i, $\mathbf{W}^{(r)}$ is a learnable weight matrix, and α_{ij} is a normalized edge weight:

$$\alpha_{ij} = \frac{\exp(\mathrm{sim}(\mathbf{h}_i^{(r)}, \mathbf{h}_j^{(r)}))}{\sum_{k\in\mathcal{N}(i)} \exp(\mathrm{sim}(\mathbf{h}_i^{(r)}, \mathbf{h}_k^{(r)}))}, \tag{3}$$

with $\mathrm{sim}(\cdot,\cdot)$ computed via dot-product, bilinear function, or a small MLP, and $\sigma(\cdot)$ a nonlinear activation such as ReLU.

 doi:10.20944/preprints202601.1559.v1

After R layers, node embeddings are aggregated into a fixed-dimensional user representation:

$$\mathbf{h} = \mathrm{AGG}\Big(\{\mathbf{h}_i^{(R)} \mid v_i \in \mathcal{V}\}\Big), \tag{4}$$

where $\mathrm{AGG}(\cdot)$ can be average pooling, attention-based aggregation, or other functions.

Graph-based methods naturally capture heterogeneous behaviors and complex dependencies among interactions [55,56], offering flexibility in modeling rich user histories. However, constructing and updating large-scale dynamic graphs for ultra-long sequences incurs substantial computational and maintenance costs, making their direct application to industrial ULBM challenging due to latency and scalability constraints [57,58].

2.3.4. Search-based and Compression-based Methods

To address the scalability challenges of ultra-long behavior sequences, search-based and compression-based methods are widely adopted as core preprocessing strategies in industrial ULBM systems. Following a *preprocess first, encode later* paradigm, these methods first reduce the effective sequence scale, either by selecting a refined subsequence $\mathbf{S}^*$ through search or by aggregating the sequence into a set of user interests $\mathcal{S}_{int}$ via compression. The subsequence or interest set is then encoded by subsequent modeling to produce the final user representation $\mathbf{h}$. By explicitly reducing the input size before downstream modeling, these methods directly target the efficiency–effectiveness balance (EEB) and have become the default choice in large-scale industrial deployments.

Building on this framework, the following sections review representative ULBM methods, emphasizing how real-world systems navigate trade-offs between modeling effectiveness and computational cost.

3. Efficiency Optimizations for ULBM

In this section, we review efficiency optimization strategies for user lifelong behavior modeling (ULBM). In industrial-scale recommender systems, user behavior sequences often contain tens of thousands of interactions or more, making direct modeling impractical under strict latency and resource constraints. To address this challenge, prior work improves efficiency from two complementary perspectives: **Algorithmic Optimizations** and **Infrastructure Optimizations**. From an efficiency–effectiveness balance (EEB) perspective, algorithmic optimizations reduce the effective input scale of the interest encoder through sequence preprocessing, while infrastructure optimizations lower the amortized computational overhead via system-level designs, enabling stable and reliable online serving. Adopting this perspective allows us to systematically summarize existing efforts and clarify how different approaches support practical industrial deployment.

3.1. Algorithmic Optimizations

We present a taxonomy of algorithmic optimizations consisting of two major families: *search-based methods* and *compression-based methods*. We examine their core design principles and methodological evolution, followed by a summary and discussion of future research directions. Unlike prior surveys that categorize methods by model architectures, this taxonomy emphasizes where and how sequence length is reduced, which directly reflects efficiency constraints in industrial deployment. We begin with search-based methods, which represent the most widely adopted ULBM strategy in industrial-scale recommender systems.

3.1.1. Search-Based Methods

These methods aim to preprocess the user lifelong behavior sequence by identifying a small subset of interactions that are most relevant to the current recommendation condition, thereby substantially reducing the effective input scale for downstream modeling. This design is motivated by the observation that not all historical behaviors contribute equally under strict efficiency constraints. Search-based methods were first introduced by SIM [42] in 2020, enabling user sequences containing up to 54,000

 doi:10.20944/preprints202601.1559.v1

historical interactions to be efficiently processed. Since then, this paradigm has been rapidly adopted in large-scale industrial recommender systems, including Tencent [59], ByteDance [60], Pinterest [61], and Meta [6]. Following the paradigm introduced by SIM, search-based methods generally consist of two components: the **Global Search Unit** (GSU) and the **Exact Search Unit** (ESU). GSU selects the top-k interactions most relevant to the specified condition from the original long sequence, producing a refined subsequence with significantly reduced length. ESU then performs fine-grained interest modeling over this subsequence to generate user representations for ranking.

Formally, given a user lifelong behavior sequence $\mathbf{S} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L]$, a search condition c, a search function f_{GSU}, and an encoding function f_{ESU}, search-based methods can be expressed as:

$$\mathbf{h} = f_{\text{ESU}}\big(f_{\text{GSU}}(\mathbf{S}, c; \boldsymbol{\theta}_G), c; \boldsymbol{\theta}_E\big), \tag{5}$$

where $\mathbf{h}$ denotes the user interest representation, and $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_E$ denote the parameter sets of f_{GSU} and f_{ESU}, respectively.

In this formulation, ***GSU is primarily optimized to improve search reliability under strict efficiency constraints***, whereas ESU focuses on enhancing the expressiveness of user interest modeling. As ESU will be discussed in detail in Sec. 4, this subsection focuses on the algorithmic design and evolution of GSU. Table 2 summarizes representative search-based methods, categorized by their search condition and search function. This taxonomy highlights how different design choices affect the balance between filtering reliability and computational efficiency at the search stage. We next elaborate on these two dimensions, starting with the search condition.

Search Condition provides target-related signals to guide the filtering of historical interactions, ensuring that the retained subsequence remains aligned with the downstream prediction objective. Most existing studies adopt the **target item** as the condition, selecting historical behaviors that are most likely to influence the user's current decision [62–64]. For instance, LCN [59] performs multi-level search using the target item, progressively refining the candidate subsequence.

In certain application scenarios, **contextual information** is also employed as the search condition. For example, CoFARS [65], designed for instant delivery services, incorporates contextual factors such as distance and time, selecting historical interactions under similar contexts. This design improves filtering precision while preserving the efficiency advantages of search-based preprocessing.

From an EEB perspective, conditioning on target items or contextual information focuses the search on interactions with higher predictive utility, increasing the effectiveness of search-based preprocessing without increasing computational cost, while the specific mechanism for approximating relevance is determined by the search function.

Table 2. A Taxonomy of Representative Search-based Methods.

Search Function \ Search Condition	Target Item	Contextual Info.
Rule-based	[42], [66]	[67], [65]
Vector Similarity	[59], [64], [61]	–
SimHash-based	[68], [69], [70]	–
Target Attention	[33], [71], [60], [72]	–

Search Function is the core component of GSU, determining how relevance between the condition and historical interactions is approximated during large-scale filtering.

Importantly, the goal of the search function is not to perform fine-grained interest modeling, but to **maximize information preservation per unit of computation** when operating over ultra-long sequences. This objective has motivated a line of work that continuously improves search efficiency, allowing increasingly long behavior histories to be pruned into compact subsequences under strict computational constraints.

 doi:10.20944/preprints202601.1559.v1

Early methods adopted **rule-based** strategies based on explicit feature matching. For example, SIM selects historical interactions whose *category-id* matches that of the target item:

$$r_i = \text{Sign}(C_i, C_t), \tag{6}$$

where r_i denotes the relevance score, C_i and C_t correspond to the category-ids of the i-th historical interaction and the target item, respectively. $\text{Sign}(\cdot,\cdot)$ is an indicator function that returns 1 if the two inputs match and 0 otherwise. While rule-based matching is computationally efficient, it captures only coarse feature-level consistency and lacks semantic generalization, which limits its filtering reliability in complex scenarios.

To improve generalization under similar efficiency constraints, subsequent studies explored **vector similarity**-based search, such as inner-product or cosine similarity in the embedding space. For instance, TransAct V2 [61] searches the top-k most relevant behaviors based on inner-product similarity with the target item embedding $\mathbf{x}_t$:

$$\mathbf{S}^* = \left\{ \mathbf{b}_i \;\middle|\; i \in \text{TopK}\left(\{\langle \mathbf{b}_j, \mathbf{x}_t \rangle\}_{j=1}^{L}, k\right) \right\}. \tag{7}$$

Here, TopK selects the indices of the top-k values.

Parameterized similarity functions extend basic inner-product computation by introducing learnable matrices to better capture semantic relationships between the target and historical items:

$$r_i = (\mathbf{W}_t \mathbf{x}_t)(\mathbf{W}_i \mathbf{b}_i)^\top, \tag{8}$$

where $\mathbf{W}_t$ and $\mathbf{W}_i$ are learnable projection matrices for the target and historical items, respectively, and r_i denotes the relevance score.

To move beyond isolated optimization of individual stages and toward joint optimization across the pipeline, ETA [68] adopts a **SimHash-based** strategy, encoding continuous embeddings into compact binary codes:

$$\mathbf{S}^{(\text{hash})} = \text{Sign}(\mathbf{S} \cdot \mathbf{H}), \quad \mathbf{x}_t^{(\text{hash})} = \text{Sign}(\mathbf{x}_t \cdot \mathbf{H}), \tag{9}$$

where $\mathbf{H} \in \mathbb{R}^{d \times m}$ is the hash projection matrix, $\text{Sign}(\cdot)$ returns 1 if the input is positive and 0 otherwise, and both $\mathbf{S}$ and $\mathbf{x}_t$ are encoded into the same hash space. By leveraging bitwise operations, the Hamming distance between each token in $\mathbf{S}^{(\text{hash})}$ and the target embedding $\mathbf{x}_t^{(\text{hash})}$ can be efficiently computed. This enables similarity computation over long sequences with $O(L)$ complexity, making joint optimization between GSU and downstream modules feasible at industrial scale, and substantially improving search efficiency. Subsequent works such as SDIM [69] and MIRRN [70] further refine this approach.

As search-based methods evolved from rule-based filtering to semantic-aware and end-to-end optimized search, the filtering reliability of GSU has been substantially improved. Nevertheless, GSU remains structurally constrained by its efficiency-oriented objective, which may lead to misalignment with the fine-grained attention mechanisms employed in ESU.

To reduce this discrepancy, TWIN [33] extends **target attention** to GSU through algorithmic–infrastructure co-design. By separating intrinsic and interaction features and introducing caching and dimensionality reduction strategies, TWIN significantly reduces the computational cost of attention-based search, making it feasible at the search stage. Details of the underlying caching mechanisms are discussed in Sec. 3.2.3. This design addresses the inconsistency between the optimization objectives of GSU and ESU, enabling more reliable filtering under fixed computational budgets, and has been adopted by subsequent methods such as TWIN V2 [31], LIC [60], and MARM [72].

Finally, recognizing that GSU and ESU exhibit gradient conflicts, DARE [71] decouples their embedding spaces to alleviate this issue, further stabilizing large-scale training without increasing inference cost.

 doi:10.20944/preprints202601.1559.v1

From an EEB perspective, search-based methods provide a practical solution for handling ultra-long user behavior sequences by reducing computational cost, thereby helping maintain the ROI equilibrium even as sequence length grows. The GSU aggressively prunes the original sequence by selecting a small set of interactions most relevant to the current target or context, enabling scalable inference even when the raw sequence spans tens or hundreds of thousands of behaviors. However, this efficiency comes at the cost of effectiveness: the search process typically retains a fixed number of interactions, discarding temporal continuity and higher-order dependencies among behaviors, and is often item-centric, limiting its ability to capture complex preference patterns or heterogeneous behavior interactions. As a result, search-based methods occupy a regime on the efficiency–effectiveness frontier that favors extreme efficiency and target specificity, while sacrificing holistic interest modeling. These limitations motivate approaches that aim to preserve richer user interest representations under comparable efficiency constraints.

3.1.2. Compression-Based Methods

Table 3. A Taxonomy of Representative Compression-based Methods.

Reduction Function \ Partition Strategy	Rule-based	Cluster-based	Direct Compress
Pooling-based	–	[31],[73]	–
Attention-based	[74], [75]	[76], [77]	[78], [37]
Discrete Codebook	–	–	[79], [80]
Transformer-based	[81], [6]	[82]	–

These methods aim to distill the user lifelong behavior sequence into a compact set of interest vectors, as directly modeling extremely long sequences is computationally costly and prone to noise. During inference, the model can prioritize the most relevant vector, reducing computation while preserving predictive effectiveness.

Motivated by this insight, compression-based methods have emerged in prior work. In this survey, we explicitly modularize this family of methods into two components: the **Interest Reduction Unit** (IRU) and the **Interest Fusion Unit** (IFU). IRU partitions the original sequence and generates a set of corresponding interest representations. IFU then integrates all representations within this set into a unified user interest representation used by the ranking model.

Formally, given a user lifelong behavior sequence $\mathbf{S} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L]$, a partition strategy π, a reduction function f_{IRU}, a encoding function f_{IFU}, compression-based methods can be expressed as:

$$\mathbf{h} = f_{\text{IFU}}\Big(f_{\text{IRU}}(\mathbf{S}, \pi; \boldsymbol{\theta}_R); \boldsymbol{\theta}_F\Big), \tag{10}$$

where $\mathbf{h}$ is the user interest representation.

In this formulation, ***IRU is primarily optimized to improve compression fidelity under strict efficiency constraints,*** whereas IFU focuses on enhancing the effectiveness of integrated user interest representations. Table 3 summarizes representative compression-based methods, organized by their partition strategy and reduction function. This taxonomy illustrates how different design choices impact the trade-off between sequence compression efficiency and the fidelity of interest representation at the compression stage. We next elaborate on these two dimensions, starting with the partition strategy.

Partition Strategy determines how an ultra-long behavior sequence is decomposed into information blocks, with the primary goal of reducing computational cost for downstream modeling. It directly shapes the efficiency–effectiveness trade-off: coarser partitions accelerate processing but may lose fine-grained user interests, whereas finer partitions better preserve semantics at higher computational cost. While a few methods implicitly reduce sequence scale via dimensionality reduction within the Reduction Function, most adopt explicit partitioning, which can be broadly categorized as rule-based or clustering-based strategy.

 doi:10.20944/preprints202601.1559.v1

In industrial practice, **rule-based** partitioning typically leverages business priors. For example, DGIN [83] partitions sequences by item categories, UUM [75] by application surfaces, and HiT-LBM [84] uses temporal order for chronological segmentation, ensuring interpretable partitioning. Furthermore, Climber [81] explicitly exploits heterogeneous user behaviors, such as click, like, and forward, to partition the original sequence. This design well aligns with the heterogeneous nature of user lifelong behavior sequences, enabling different behavior types to be explicitly modeled.

In contrast, **clustering-based** partitioning employs data-driven algorithms to group similar behaviors, aiming to reduce sequence length under efficiency constraints while preserving latent semantic structures. For example, ENCODE [76] applies linear dimensionality reduction followed by K-Means clustering over full behavior sequences. DMGIN [82] leverages a CLIP-based [85] model to extract image–text embeddings and cluster items with similar semantics. Trinity [73] maps items to cluster centroids and maintains real-time histograms over cluster IDs to capture evolving user interests.

From an EEB perspective, partitioning transforms the original sequence into a set of manageable blocks, reducing the input scale for downstream modeling while retaining a more comprehensive view of the original sequence, with the specific modeling of user interest within each block performed separately by the reduction function.

Reduction Function determines how much information is retained from each block and in what form, directly shaping the trade-off between information preservation and computational efficiency. Across existing methods, interest reduction has progressively evolved from simple continuous pooling toward more structured mechanisms, including attention-based selection, discrete codebook-based compression, and Transformer-integrated reduction, aiming to preserve informative signals under strict efficiency constraints.

The naive designs adopt **pooling-based** reduction over each subsequence. For example, TWIN V2 [31] further tighten the coupling between partitioning and reduction by using cluster-centroid representations. Each cluster is treated as a virtual item and can be represented either by the real behavior closest to the centroid or by the mean feature of all items in the cluster. Then the aggregated feature is mapped through an embedding layer into a compact vector. This approach offers extremely low-cost per-block reduction and is easy to deploy at scale, but since the pooling scheme treats all behaviors within a block as equally important, it cannot adapt the representation to different targets or contexts.

To increase adaptability, **attention-weighted** reduction learns the importance of each interaction. ULIM uses a short-term interest vector $\mathbf{h}_{\text{short}}$ as a query to compute target-aware attention over each category-specific subsequence. This process can be formally expressed as:

$$\mathbf{h}_i = \mathrm{TA}\big(\mathbf{h}_{\text{short}}, \mathbf{S}_i\big), \tag{11}$$

where $\mathbf{h}_i$ denotes the interest representation of the i-th subsequence $\mathbf{S}_i$ and $\mathrm{TA}(\cdot)$ implements target attention. ENCODE similarly applies target attention within each cluster, using the cluster centroid as the query to reweight interactions inside the cluster. From this perspective, attention-based reduction allows the same subsequence to yield different vectors under varying targets, capturing both contextual and semantic heterogeneity at moderate cost.

As user sequences grow longer, the number of unique items increases, expanding the embedding table and raising memory demands. **Discrete codebook-based** reduction addresses this by quantizing the original sequence into a fixed set of learned codes, reducing memory usage while preserving semantic relationships. For example, VQL [79] adopts learned codebooks to represent sequences with a fixed number of discrete codes, and CHIME [80] uses multi-level residual codebook quantization:

$$\begin{aligned} k^{(m)} &= \arg\min_{k \in [K_m]} \left\| \mathbf{e}^{(m)} - C_k^{(m)} \right\|_2, \\ \mathbf{c}^{(m)} &= C_{k^{(m)}}^{(m)}, \\ \mathbf{e}^{(m+1)} &= \mathbf{e}^{(m)} - \mathbf{c}^{(m)}, \end{aligned} \tag{12}$$

 doi:10.20944/preprints202601.1559.v1

where $C^{(m)} \in \mathbb{R}^{K_m \times D_m}$ denotes the codebook at layer m, $\mathbf{e}^{(m)}$ is the input to this layer, $k^{(m)}$ is the selected code index, and $\mathbf{c}^{(m)}$ is the corresponding code vector. The code indices are then aggregated into histograms, forming compact representations for downstream ranking models. In practice, discrete codebook–based methods avoid explicit sequence partitioning, efficiently compress sequences, and capture latent heterogeneous semantics, enabling effective interest representation under computational and memory constraints.

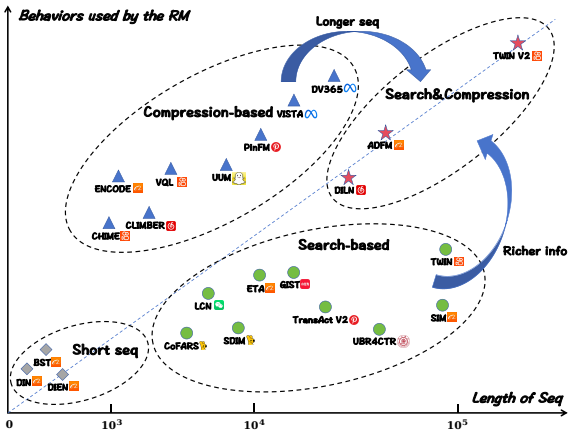

Figure 5. Trend chart of existing ULBM work on algorithmic optimizations. The positions of individual works are qualitative rather than quantitative.

A line of work applies **Transformer-based** reduction to subsequences. DV365 [6] employs a Funnel Transformer [86] with stacked blocks, applying strided mean pooling along the token dimension at each layer to progressively compress user interest representations. Climber [81] uses an Adaptive Transformer to encode subsequences extracted under different strategies π_i and recommendation scenarios r, producing dedicated representations:

$$\mathbf{h}_i = \text{Transformer}(\mathbf{S}_i; \pi_i, r), \tag{13}$$

where π_i and r provide bias and normalization for attention computation. Although Transformer-based designs are computationally heavier, efficiency-oriented modifications, such as strided pooling and pre-extracted subsequences, support their application under strict computational constraints, enabling scalable modeling of user interests while balancing representation fidelity and efficiency.

From an EEB perspective, compression-based methods improve scalability by transforming the full lifelong behavior sequence into a compact set of reusable interest representations through the IRU. By compressing the user sequence once and reusing the resulting interest representations across multiple candidate items, IRU allows downstream models to exploit richer historical information under bounded computational cost, thereby helping maintain the ROI equilibrium even as sequence length grows. Nevertheless, this design introduces clear effectiveness trade-offs. Since IRU is typically target-agnostic, interactions irrelevant to a specific recommendation can be compressed into the same representation, which may introduce noise into subsequent modeling. In addition, interest representations are often updated at coarse time granularity to enable reuse, resulting in a mismatch between the compressed interests and the user's most recent behaviors. Consequently, compression-based methods emphasize efficiency through reuse and abstraction, at the expense of target specificity and temporal freshness, positioning them at a complementary point on the efficiency–effectiveness frontier relative to search-based approaches.

3.1.3. Summary

Search-based and compression-based methods leverage the Global Search Unit (GSU) and the Interest Reduction Unit (IRU) to preprocess ultra-long user lifelong behavior sequences, improving ULBM efficiency by reducing the input to the Interest Encoder. From an EEB perspective, this two-

 doi:10.20944/preprints202601.1559.v1

stage paradigm performs a structured transformation of the original sequence, explicitly trading computational cost against the fidelity of interest representations.

Figure 5 provides a qualitative overview of existing ULBM studies, showing how different approaches utilize user behavior information in ranking models as sequence length increases. In the figure, each circle represents a specific ULBM work, positioned according to the length of user behavior sequences (horizontal axis) and the number of behaviors leveraged by the ranking model (vertical axis). As illustrated, search-based GSU prioritizes selectivity under extreme sequence lengths, allocating limited computation to a small subset of interactions with the highest immediate relevance. This design makes it particularly effective for ultra-long histories, but its candidate-dependent nature ties efficiency gains to repeated search operations and restricts the amount of contextual information available to downstream modeling.

By contrast, compression-based IRU emphasizes reuse and abstraction by distilling the entire behavior history into compact interest representations that can be shared across candidates. This allows downstream models to access broader historical signals under stable computational cost and facilitates the incorporation of heterogeneous behavior types. The resulting efficiency, however, comes at the cost of reduced target specificity, as compressed representations may blend relevant and irrelevant signals and lag behind users' most recent behaviors.

These complementary trade-offs suggest that neither selectivity nor abstraction alone is sufficient. To leverage both longer sequences and richer information, **hybrid methods** integrate search-based and compression-based designs to jointly balance efficiency and effectiveness. DILN [66] first selects relevant interactions per action type via rule-based matching and inner-product similarity, then compresses these subsequences with a CNN [87] to obtain the interest representation $\mathbf{h}$. TWIN V2 [31] compresses the user sequence into latent clusters and applies cluster-aware target attention to compute $\mathbf{h}$. In industrial-scale recommender systems, preprocessing pipelines may follow either a *search first, compression later* [66] or *compression first, search later* [31,82,88,89] strategy. Integrating these paradigms leverages their complementary strengths, making hybrid preprocessing a promising direction that efficiently balances computational cost and rich user interest representation, consistent with EEB objectives.

3.2. Infrastructure Optimizations

Beyond algorithmic optimizations, infrastructure-level enhancements are essential to meet the latency and throughput demands of industrial ULBM deployments. Consequently, infrastructure optimizations become the key pathway for practical system deployment. Unlike general recommender models, ULBM uniquely stresses infrastructure due to ultra-long sequences, persistent user states, and frequent online inference. The central objectives of these optimizations are to increase inference throughput, reduce GPU memory and storage overhead, and shorten the overall response time. Compared with innovations in previous subsection, infrastructure optimizations focus more on the organization and utilization efficiency of computational resources. Figure 6 presents a hierarchical taxonomy of mainstream infrastructure optimization techniques. Specifically, these techniques can be categorized into three groups: custom kernel optimization, precision optimization, and multi-level caching mechanisms. These techniques target distinct system-level performance bottlenecks. Custom kernel improves execution efficiency at the kernel and computation-graph levels. Precision optimization seeks an effective balance between numerical storage costs and computation efficiency under reduced-precision constraints. Multi-level cache mechanisms realize structured reuse by trading storage for computation. Collectively, these techniques fundamentally reshape the execution cost profile of ULBM, without altering their underlying modeling objectives. We will now delve into each category of infrastructure optimizations, beginning with custom kernel design, followed by precision optimization and multi-level cache mechanisms.

 doi:10.20944/preprints202601.1559.v1

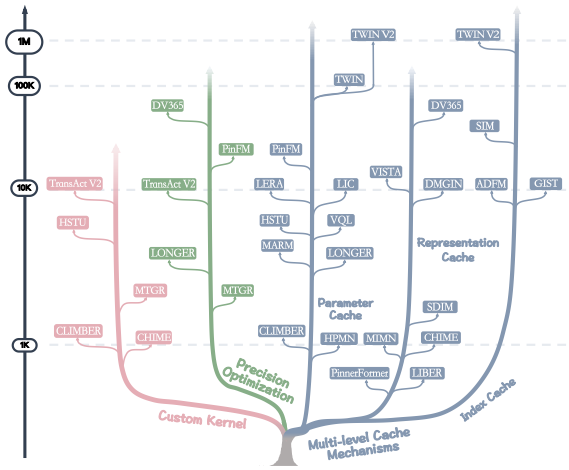

Figure 6. The infrastructure optimizations of ULBM can be organized along three dimensions: custom kernel design, precision optimization, and multi-level cache mechanisms.

3.2.1. Custom Kernel

Custom kernel is particularly important for user lifelong behavior modeling, as attention mechanisms are often applied to ultra-long sequences, requiring frequent kernel scheduling. To address this, kernel-level optimization primarily aims to eliminate redundant data movement and reduce scheduling overhead. In industrial practice, two strategies are commonly adopted: (I) designing custom GPU kernels to replace frequently invoked general-purpose kernels; (II) applying kernel fusion to reduce memory I/O and the frequency of kernel launches (such as fusing QKV projection with FlashAttention or embedding lookup with attention computation).

TransAct V2 [61], built on the Triton compiler framework, exemplifies industrial kernel optimization by implementing a Single Kernel Unified Transformer (SKUT) architecture that executes QKV projection, FlashAttention [90], LayerNorm, and FFN layers cohesively within on-chip SRAM. This design avoids intermediate tensor materialization and multiple kernel invocations, yielding a 6.6× forward-pass speedup over PyTorch, 85% latency reduction, and substantial GPU memory savings. HSTU [91] further demonstrates attention-level optimization via sparsified and grouped-GEMM [92] pipelines with fused relative positional bias, achieving 5.3–15.2× speedups compared to FlashAttention V2 [93]. MTGR [94] builds upon HSTU with FP16 computation and careful kernel-level load balancing, providing additional 1.6–2.4× throughput improvements in heterogeneous sequence workloads.

Other industrial systems, such as CHIME [80] and Climber [81], apply FlashAttention-style tiling and kernel fusion to unify embedding lookup with attention computation, optimizing memory access and parallelism, achieving moderate but consistent speedups.

Overall, industrial-scale recommender systems rely on deeply customized kernels, making unified benchmarking challenging. Nonetheless, a clear trend emerges: kernel-level optimization consistently focuses on fusing computation paths, reducing GPU memory access, and minimizing scheduling overhead, providing the most direct acceleration for ULBM. From an EEB perspective, these efficiency gains come at the cost of reduced portability and higher engineering complexity, due to tight coupling with specific hardware and execution environments, rather than changes in model expressiveness. Nonetheless, kernel-level optimization remains a critical engineering pillar for high-throughput ULBM inference.

3.2.2. Precision Optimization

Precision optimization is particularly vital for ULBM, where massive embedding tables and long-sequence computations impose substantial memory and compute pressure. Unlike conventional recommender models that operate on short behavior windows, ULBM models extremely long user histories, which substantially expands the effective item coverage and long-tail exposure that must be supported by the system. Moreover, long-context attention further amplifies resource consumption: the dimensionality of QKV projections and intermediate activations scales linearly with sequence

 doi:10.20944/preprints202601.1559.v1

length, and these tensors are repeatedly accessed during inference. As a result, memory footprint, bandwidth demand, and arithmetic intensity all increase significantly, making reduced-precision storage and computation a critical enabler for practical ULBM deployment. To mitigate these costs, two complementary strategies are commonly adopted: (I) reducing storage overhead through lower bit-width representations, such as INT8/INT4 quantization of embeddings; (II) improving compute utilization via mixed-precision execution, for example, maintaining critical layers in FP16 while auxiliary layers operate at lower precision.

The first strategy is exemplified by TransAct V2 [61], which introduces INT8 quantization into multi-stage retrieval and long-sequence modeling pipelines. Core embeddings are converted from FP16 to INT8 at storage time, for deployment, substantially reducing memory usage, while dequantization combined with kernel fusion during inference achieves latency reduction with negligible accuracy degradation. Along similar lines, DV365 [6] further integrates INT4 quantization with caching mechanisms, enabling efficient GPU-side reuse of quantized embeddings while preserving full-precision computation during training.

In contrast, mixed-precision strategies focus on improving computational efficiency by selectively assigning numerical precision to different model components. LONGER [95] exemplifies this direction by maintaining critical layers in FP16 to ensure training stability, while allowing auxiliary modules to operate at lower precision, thereby balancing numerical robustness and computational efficiency. Related efforts, such as PinFM [96] and MTGR [94], combine mixed-precision execution with customized kernels, coupling precision reduction with kernel-level acceleration to improve throughput.

Fundamentally, precision optimization mitigates the memory and bandwidth bottlenecks arising from embedding storage, KV-cache management, and attention computation in ULBM by eliminating redundant representations, compressing data pathways, and improving compute utilization. Rather than functioning in isolation, precision optimization operates in concert with custom kernel and caching mechanisms, enabling ULBM systems to preserve representational capacity and stability at significantly lower resource cost in large-scale industrial deployments. From an EEB perspective, these efficiency gains are achieved with minimal impact on modeling effectiveness, illustrating a careful balance between computational efficiency and representational fidelity in large-scale industrial deployments.

Table 4. Summary of Representation Cache.

Model	Original Seq. Length	# Representations	Dimensions
MIMN	1,000	2*m, m=4/6/8	16
CHIME	1,000	1	64
SDIM	[1000, 2000]	16	128
DMGIN	10,000	50	16
DV365	avg. 40,000	58	256
VISTA	12,000	[100, 1000]	256

3.2.3. Multi-level Cache Mechanisms

Cache mechanisms represent a critical pathway for enhancing the efficiency of ULBM in industrial-scale recommender systems. Due to the extremely high computational cost of processing ultra-long user sequences, real-time inference would be prohibitively expensive. By strategically storing intermediate representations, model parameters, or relative positional indices, cache leverages additional storage to reduce computation, while the relatively slow evolution of long-term user interests ensures that cached representations remain valid for repeated reuse.

Representation cache is a representative form of cache mechanisms, designed to reuse intermediate or final representations within the user behavior modeling pipeline, thereby reducing the computational cost of online inference.

Several early approaches cache intermediate results or modeling units to alleviate the computational burden of long-sequence modeling. MIMN [24] adopts a memory network that stores a fixed set

 doi:10.20944/preprints202601.1559.v1

of memory slots and internal states as a persistent cache, thereby avoiding repeated computation over extended behavior histories during inference. SDIM [69] applies hashing to long user sequences and organizes the resulting hash buckets as reusable cached descriptors, enabling efficient retrieval and reuse of compressed behavioral representations.

As user behavior sequences continue to grow longer, many ULBM methods follow a *preprocess first, encode later* paradigm. In such settings, representation compression is commonly applied to directly store the outputs of the preprocessing stage, enabling efficient reuse during encoding. In large-scale industrial recommender systems, representation cache further enables a complete decoupling between preprocessing and encoding, allowing cached user representations to be loaded with $O(1)$ complexity during online inference and thereby dramatically reducing latency.

LIBER [97] exemplifies this by generating user representations via large language models and deploying them with partitioning and cache strategies, substantially reducing inference latency. CHIME [80] and DV365 [6] also cache compressed user embeddings or interest representations to support multiple downstream tasks efficiently. VISTA [37], DMQN [77], and DMGIN [82] follow similar principles and also cache summarized or HSTU-processed outputs for fast online serving.

Table 4 presents several works employing representation cache, showing the original sequence lengths and the number of representations after compression. These methods generally achieve a hundredfold reduction, substantially lowering the service latency of the online inference stage. In most cases, representation cache is applied to store the outputs of the preprocessing stage in compression-based methods, as the resulting user interest representations are independent of candidate items. This allows cached representations to be reused across multiple candidate items during online inference, substantially improving serving efficiency. However, from an EEB perspective, representation cache also introduces limitations: cached interests may become stale due to delayed updates, and cold-start users are not well supported.

Beyond representation cache, another important form is **parameter cache**, which aims to store parameters used in ULBM to reduce computational overhead. Essentially, the cached parameters are candidate-independent and can be computed once and reused multiple times without being affected by the candidate item, such as weight matrices $\mathbf{W}$ and the Key/Value components in attention mechanisms. By enabling reuse through parameter cache, part of the computational cost is shifted to memory overhead, substantially reducing forward computation during inference and alleviating latency bottlenecks in ULBM [98].

Several ULBM methods leverage parameter cache to accelerate inference. TWIN [33] replaces real-time linear transformations with offline projection and KV cache, refreshing non-sequential features every 15 minutes while pruning long-tail users. LIC [60] applies a similar offline projection and cache strategy for Q/K-side features.

Hierarchical KV cache is employed in systems such as PinFM [96], LONGER [95], and Climber [81], where first-level caches store Key/Value pairs for user context and second-level caches store cross-attention activation maps, enabling reuse across candidate-item computations. MARM [72] extends this idea by storing intermediate embeddings from multi-layer Transformers with hashed keys and replacing full masked self-attention with lightweight target-attention for fast incremental updates.

Other approaches, such as LERA [99] and VQL [79], introduce specialized memory structures: LERA writes compressed linear mappings into High Bandwidth Memory [100] for direct reuse during inference, while VQL maintains a three-level V-cache based on quantized K vectors to balance latency and storage overhead.

Table 5 presents several works employing parameter cache. All of these methods utilize cache during inference, yet their inference complexity remains directly or indirectly dependent (e.g., VQL, LERA) on the sequence length L. This indicates that parameter cache cannot achieve full decoupling of training and inference like representation cache, as computation over the original sequence is still required. Generally, parameter cache is commonly applied in search-based methods, since the subsequence selection during preprocessing is highly dependent on the candidate item, making it

 doi:10.20944/preprints202601.1559.v1

possible to cache only candidate-independent parameters. From an EEB perspective, parameter cache provides moderate efficiency gains while maintaining full modeling effectiveness, although these gains come with increased engineering complexity.

Table 5. Summary of Parameter Cache.

Model	Cache Type	Used in Training	Train Complexity	Infer Complexity
TWIN	K	✓	$O(BLd + Bkd)$	$O(BLd + Bkd)$
MARM	KV	✓	$O(BLd * N)$	$O(BLd * N)$
PinFM	KV	✓	$O(BLd/c)$	$O(BLd)$
LONGER	KV	✗	$O(BL^2d)$	$O(BLd)$
Climber	KV	✓	$O(Bk^2d)$	$O(Bk^2d)$
VQL	V	✗	$O(BLkd + Bkd)$	$O(Bkd)$
LIC	QK	✓	$O(BLd + Bkd)$	$O(BLd + Bkd)$
LERA	KV	✗	$O(BLd * r + Brd)$	$O(Brd)$

Here, N is the number of layers in MARM, c is the aggregation granularity (set to 16 in PinFM), and r is the rank of the low-rank matrices in LERA.

Another complementary caching strategy is the **index cache**, designed for search-based ULBM where the final user representation depends on candidate items, making standard representation caching ineffective. Instead, index cache stores search subsequence indices or candidate sets, reducing the computational cost of online matching while accommodating candidate-dependent computations.

This line of work was first systematically developed in the SIM [42] framework, which introduced the two-stage search paradigm. Industrial deployments of SIM typically construct a User Behavior Tree that follows the Key-Key-Value data structure and enables fast search by organizing indices according to SIM's category-based matching preprocessing strategy. TWIN V2 [31] extends index cache to million-scale user lifelong behavior sequences. It reduces candidate set size via hierarchical clustering compression. Items are first clustered and then organized into an index tree, enabling logarithmic-time search and reducing inference latency. ADFM [88] pre-stores the top-k user behaviors selected by the Behavior Selection Unit, and reuses them during inference. Similarly, GIST [64] caches top-k items from the source domain and transfers them to the target domain, demonstrating the potential of index cache in cross-domain and multimodal recommendation scenarios.

In some cases, index cache serves as an optimization of parameter cache (e.g., TWIN V2), reducing online inference complexity from linear in L to logarithmic. This enables the candidate-dependent General Search Unit (GSU) in search-based ULBM methods to efficiently handle sequences with millions of interactions during online inference. From an EEB perspective, index cache improves efficiency for ultra-long sequences by storing precomputed item indices, reducing online computational cost.

3.2.4. Summary

Infrastructure optimizations form the performance foundation for deploying ULBM in industrial-scale recommender systems. The substantial computational and memory pressure introduced by ULBM becomes the dominant bottleneck: ultra-long user behavior sequences cause extremely high computational complexity, parameter scale, and memory traffic to grow multiplicatively, making it difficult for conventional computing resources to support them directly. Basic optimization strategies, including kernel fusion combined with reduced-precision computation, can alleviate part of the computational burden and reduce memory time, which corresponds to bandwidth consumption. More customized solutions, such as KV cache and related cache mechanisms, leverage the structural characteristics of user behavior sequences to shift a large portion of computing time into more controllable memory time, yielding far larger gains than generic kernel-level or precision-level optimization.

Overall, custom kernel, precision optimization and cache mechanisms jointly serve as the performance pillars for ULBM deployment. Industrial recommender systems typically adopt a co-design approach that integrates these techniques. Among them, cache mechanisms have emerged as the key enabler for the practical and efficient deployment of ULBM in industrial environments. From an EEB perspective, infrastructure optimizations primarily aim to improve efficiency, generally without

 doi:10.20944/preprints202601.1559.v1

altering model expressiveness. However, certain mechanisms such as cache may introduce slight deviations in performance. These optimizations operate in close synergy with algorithmic design to maintain a balance between computational efficiency and modeling effectiveness, thereby sustaining ROI under the growing scale of user behavior sequences.

3.3. Discussion

Algorithmic strategy sets the ceiling for infrastructure efficiency. Search-based methods rely on candidate items to derive user-specific subsequences, whereas compression-based methods generate candidate-independent user interest representations. No matter how advanced kernels, precision, or cache optimizations are, they cannot surpass the limits imposed by sequence reduction and representation design. Within this constraint, algorithm and infrastructure must be co-designed to achieve practical efficiency gains, highlighting that modern large-scale ULBM models are deeply integrated systems where algorithmic strategies (e.g., two-stage paradigm) and infrastructure solutions (e.g., multi-level cache) jointly make large-scale deployment feasible.

4. Effectiveness Optimizations for ULBM

In this section, we review effectiveness optimization strategies for user lifelong behavior modeling (ULBM). We conceptualize effectiveness as the challenge of enhancing the quality of user representations by addressing three fundamental limitations inherent in ULBM. First, summarizing long-term user interactions can lead to the loss of critical dependencies, reducing the fidelity of interest modeling. Second, overlapping or entangled behavioral signals introduce noise and ambiguity, making it difficult to accurately capture individual preferences. Third, relying solely on observed behaviors imposes a semantic ceiling, as many user intentions and decisions are strongly tied to background knowledge.

Building on this conceptual framing, Figure 7 illustrates how existing methods can be organized along three complementary dimensions: **Advancing interaction modeling capability** focuses on capturing complex dependencies across extended behavior sequences. **Fine-grained user interest understanding** refines representations to disentangle overlapping signals and characterize subtle preference variations. **Incorporating worldwide knowledge** leverages external information to enrich representations with semantic context beyond observed behavior sequences. We next delve into each of these three dimensions in detail.

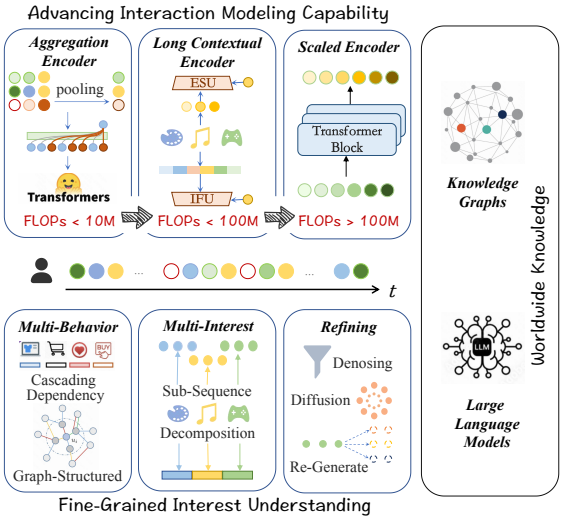

Figure 7. The effectiveness optimizations of ULBM can be characterized along three dimensions: advancing interaction modeling capability, fine-grained user interest understanding, and incorporating worldwide knowledge.

4.1. Advancing Interaction Modeling Capability

In this subsection, we introduce the Interest Encoder (IE), which generates the user interest representation **h** used by subsequent ranking modules. Depending on the processing strategy and the length of the input sequence, IEs can be categorized into three types, reflecting a progressive

 doi:10.20944/preprints202601.1559.v1

expansion in the modeling scale of the encoders. *Aggregation Encoder* handles short sequences of approximately hundreds of interactions in a single stage, with research focusing on effective token aggregation strategies. *Long Contextual Encoder* processes ultra-long sequences in a two-stage manner, explicitly balancing modeling effectiveness and computational efficiency to meet the constraints of industrial-scale recommender systems. However, recent industrial practices suggest that the optimal efficiency–effectiveness trade-off of two-stage paradigms is gradually shifting. In particular, aggressively scaling sequence length under low-capacity models leads to diminishing marginal gains, motivating alternative designs that allocate more computation to modeling richer dependencies. These observations motivate a new research direction: instead of aggressively increasing sequence length under low-capacity models, recent work increasingly focuses on exploiting moderate-length sequences with substantially increased model capacity. *Scaled Encoder* follows this philosophy by stacking computational resources and directly modeling the original user behavior sequence in a single stage, aiming to fully leverage rich temporal and semantic dependencies for ULBM. We first discuss Aggregation Encoder, which are designed for shorter sequences and represent the foundational stage in the evolution of Interest Encoders.

4.1.1. Aggregation Encoder

Early works on User Behavior Modeling (UBM) primarily focus on aggregating user historical interactions, where sequence lengths are typically limited to hundreds. GwEN [44] represents one of the earliest approaches, grouping feature embeddings and applying mean pooling to obtain user representations. DIN [45] further incorporates the candidate item into UBM by adopting an attention mechanism to model relevance between the target item and historical behaviors, followed by sum pooling. However, these aggregation-based methods ignore the temporal order of interactions, which constrains their modeling capacity.

To capture sequential dependencies, DIEN [101] introduces a GRU [102], enabling order-aware modeling at the cost of increased latency due to its inherently serial computation. BST [46] advances this line of work by adopting the Transformer architecture, which models user behavior sequences through self-attention while encoding temporal information via positional embeddings.

Given a user behavior sequence $\mathbf{S} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L]$ and a target item $\mathbf{b}_{L+1}$, BST constructs input embeddings by concatenating each behavior embedding with its positional encoding $\mathbf{p}_i$, i.e., $\mathbf{e}_i = \text{concat}(\mathbf{b}_i, \mathbf{p}_i)$. The Transformer encoder then captures sequential dependencies via self-attention and a feed-forward network f_{FFN}:

$$\mathbf{h} = \text{Transformer}\big([\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{L+1}]\big). \tag{14}$$

Overall, these methods progressively enrich user representations by incorporating candidate-aware relevance modeling and temporal dependencies. However, as user interactions continue to accumulate, sequences of only hundreds of behaviors become insufficient to characterize long-term interests, motivating the extension from short-term UBM to lifelong behavior modeling.

4.1.2. Long Contextual Encoder

When the sequence length reaches the scale of thousands or more, incorporating and exploiting longer contextual information becomes increasingly important for understanding and capturing user interests. Generally speaking, ULBM follows a *preprocess first, encode later* paradigm. The preprocessing stage primarily aims to reduce the input scale, as detailed in Sec. 3.1, while this subsection focuses on modeling high-quality user interest representations based on the results of preprocessing. For search-based methods, the IE corresponds to the **Exact Search Unit** (ESU), whereas for compression-based methods, the IE corresponds to the **Interest Fusion Unit** (IFU). This abstraction is introduced for analytical clarity rather than to propose new modules.

For search-based methods, the input to the ESU is the subsequence generated during preprocessing, denoted as $\mathbf{S}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_k^*]$, where k is typically on the order of hundreds. A common

 doi:10.20944/preprints202601.1559.v1

design adopts target attention to model the relevance between $\mathbf{S}^*$ and the candidate item $\mathbf{x}_t$. SIM [42] is among the first to employ this strategy to derive the user interest representation $\mathbf{h}$:

$$\mathbf{h} = \mathrm{TA}(\mathbf{S}^*, \mathbf{x}_t), \tag{15}$$

where $\mathrm{TA}(\cdot)$ denotes the target attention mechanism. This design is subsequently adopted by methods such as ETA [68], TWIN [33], and DARE [71]. However, these approaches primarily focus on interactions between $\mathbf{S}^*$ and the candidate item, leaving richer contextual information under-exploited.

To alleviate this limitation, recent works enhance ESU with additional contextual signals, which we unify under the term **context-aware target attention**. For example, LIC [60] incorporates temporal information by introducing clock features in both GSU and ESU, enabling temporally informed interaction modeling:

$$\begin{aligned} \alpha(\mathbf{b}_i, \mathbf{x}_t) &= \underbrace{\frac{(W_b\mathbf{b}_i) \odot (W_q\mathbf{x}_t)^\top}{\sqrt{d}}}_{\text{item similarity}} + \underbrace{s(\Delta(b_i, t))}_{\text{time similarity}}, \\ \mathbf{h} &= \mathrm{softmax}(\alpha)^\top \mathbf{Z}\mathbf{W}_V, \end{aligned} \tag{16}$$

where $\Delta(b_i, t)$ encodes temporal differences and $s(\cdot)$ is implemented as an MLP. $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots \mathbf{z}_k]$, where each $\mathbf{z}_i$ is obtained by concatenating the embedding of $\mathbf{b}_i$ with $\Delta(b_i, v)$. Beyond temporal context, GIST [64] and TWIN V2 [31] incorporate structural statistics from GSU, while MUSE [103] further integrates multimodal signals, such as textual and visual features, into the attention mechanism. Collectively, these methods demonstrate that enriching ESU with contextual cues leads to more targeted modeling and consistent performance gains in industrial settings.

Despite these improvements, context-aware target attention still inadequately captures dependencies within the subsequence itself. To address this limitation, more recent works adopt the Transformer architecture as the ESU backbone, exemplified by TransAct V2 [61]. Given the subsequence representations $\mathbf{S}^*$ and auxiliary encodings of the candidate item, positional information, and action types, a Transformer encoder produces the final user representation:

$$\begin{aligned} \mathbf{u} &= \mathrm{Transformer}(\mathbf{X}), \\ \mathbf{h} &= \big[\mathrm{Linear}(\mathbf{u}); \mathrm{MaxPool}(\mathbf{u})\big]. \end{aligned} \tag{17}$$

By modeling both sequence–item interactions and intra-sequence dependencies, Transformer-based ESU designs yield higher-quality user interest representations, often representing the upper bound of ESU effectiveness at the cost of increased computational complexity.

When we shift our focus to compression-based methods, their key distinction lies in the fact that the IRU takes as input the set of user interest representations, $\mathcal{S}_{int}$, whose size is determined during preprocessing and typically does not exceed one hundred. Unlike subsequences, elements in $\mathcal{S}_{int}$ do not exhibit strict temporal dependencies, offering greater flexibility in algorithm design. Consequently **static aggregation** strategies such as concatenation or pooling have also been widely adopted. For example, CAIN [78] directly concatenates multiple interest representations, while UUM [75] fuses them using pooling-based operations. Benefiting from the unordered nature of $\mathcal{S}_{int}$, these methods do not require additional consideration of temporal information and are therefore computationally efficient. However, such IFU implementations based on concatenation or pooling are inherently static, as they do not adapt to changes in the underlying interest representations.

In contrast, a **gating mechanism** can dynamically assign different weights to each interest vector, enabling adaptive fusion. In Climber [81], given $\mathcal{S}_{int} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ where n denotes the set size, an Adaptive Transformer Layer is employed to implement this gating mechanism. The computation can be formalized as follows:

$$\begin{aligned} \mathbf{G} &= \mathrm{ATL}\big([\mathbf{h}_1, \mathbf{h}_1, \dots, \mathbf{h}_n]\big), \\ \mathbf{h} &= \mathbf{G} \odot \sigma\big(f_{\text{gate}}(\mathbf{G})\big), \end{aligned} \tag{18}$$

 doi:10.20944/preprints202601.1559.v1

where $\odot$ denotes element-wise multiplication, σ is the sigmoid activation, and f_{gate} represents a squeeze-and-excitation module [104]. The specific implementation of ATL is given in Eq. (13). The gating mechanism enables dynamic allocation of weights to different interest representations. While this design introduces additional computational cost compared to static fusion schemes, the overhead is generally manageable in practice and leads to improved modeling flexibility and higher-quality user interest representations. However, since the candidate item is not incorporated into the interest encoding, there remains room for further improvement.

To bridge the gap between the current interest representation and the candidate item interest, **target attention** based methods have also been widely adopted. Methods such as VQL [79] and ENCODE [76] follow this strategy by explicitly and effectively incorporating the candidate item into the IFU to enable more targeted modeling.

Whether based on ESU or IFU, both approaches perform in-depth user interest modeling by combining the outputs of the preprocessing stage with richer contextual information. However, neither can directly exploit the full information contained in the original lifelong behavior sequence. This limitation reflects an inherent trade-off in the two-stage paradigm: to make modeling tractable under realistic efficiency constraints, only a carefully compressed or selected subset of interactions from sequences of tens of thousands of behaviors can be utilized, which in turn constrains further improvements in ULBM performance. Consequently, recent studies have begun to explore methods that directly model the original user lifelong behavior sequence.

4.1.3. Scaled Encoder

Instead of generating intermediate interest representations and concatenating them with auxiliary features for downstream ranking, recent approaches aim to model the original user lifelong behavior sequence in an **end-to-end** manner. In two-stage paradigms, computational resources are dispersed over extremely long sequences, resulting in limited modeling capacity per token and fragmented modeling of user behavior continuity. End-to-end approaches reallocate computational budgets by concentrating model capacity within a unified architecture, enabling more precise modeling of dependencies across user behaviors. In this paradigm, heterogeneous signals, including user profiles, item features, and contextual information, are transformed into a unified representation and jointly processed by the interest modeling network.

Since all tokens need to be modeled in a single stage, a larger model capacity is required to thoroughly capture user interaction patterns and latent preferences. One representative direction is stacking multiple attention layers. For instance, LONGER [95] treats user profiles and candidate item features as global tokens, denoted by $\mathbf{G} \in \mathbb{R}^{m \times d}$, and simultaneously extracts a subset $\mathbf{S}_{\text{samp}} \in \mathbb{R}^{k \times d}$ from the user sequence $\mathbf{S} \in \mathbb{R}^{L \times d}$ according to specific strategies, such as selecting the most recent k tokens. It first applies a cross attention layer to reduce the effective sequence size and then performs deep modeling through multiple self-attention layers. The resulting user representation $\mathbf{h}$ is directly used by the ranking model. Formally, this process can be expressed as:

$$\begin{aligned} \mathbf{Q} = [\mathbf{G}; \mathbf{S}_{\text{samp}}]\mathbf{W}_{\mathbf{Q}}, \mathbf{K} = [\mathbf{G}; \mathbf{S}]\mathbf{W}_{\mathbf{K}}, \mathbf{V} = [\mathbf{G}; \mathbf{S}]\mathbf{W}_{\mathbf{V}}, \\ \mathbf{h} = \text{SA}(\text{CA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}; \mathbf{M})), \end{aligned} \tag{19}$$

where $\mathbf{M}$ is the causal attention mask. The operator $[\cdot\,;\cdot]$ denotes row-wise concatenation. The CA and SA mechanisms correspond to cross-attention and self-attention implementations, respectively. Cross attention reduces the modeling scale from $(m + L) \times d$ to $(m + k) \times d$, where $k \ll L$. This reduction in sequence length allows the subsequent self-attention layers to efficiently capture the dependencies within the user sequence.

STCA [105] adopts stacked cross attention layers as the main modeling backbone, followed by a RankMixer [106] to enable token-level interactions, thereby supporting the modeling of ultra-long user sequences with tens of thousands of interactions within a controllable computational budget.

 doi:10.20944/preprints202601.1559.v1

Additionally, some studies have optimized the standard Transformer architecture to achieve improved performance. HSTU [91] processes all tokens in temporal order, denoted as $\mathbf{X}$. It introduces a gating vector $\mathbf{U}$ to modulate the attention-pooled values, which can be formalized as follows:

$$\begin{aligned} \mathbf{U}, \mathbf{Q}, \mathbf{K}, \mathbf{V} &= \text{Split}\big(\sigma_1(f_1(\mathbf{X}))\big), \\ \mathbf{Y} &= f_2\Big(\text{LN}\big(\sigma_2\Big(\mathbf{Q}\mathbf{K}^\top + \mathbf{rab}^{p,t}\Big)\mathbf{V}\big) \odot \mathbf{U}\Big), \end{aligned} \tag{20}$$

where $f_i(\mathbf{X})$ denotes a MLP, σ_1 and σ_2 are nonlinear activation functions such as SiLU, LN represents layer normalization, and $\mathbf{rab}^{p,t}$ denotes the relative attention bias incorporating positional and temporal information. This architecture integrates positional and temporal information and employs gating to modulate the attention-pooled outputs. Stacking multiple layers and scaling up model capacity further enhances the ability to capture complex user interest patterns. Such designs have been widely adopted in subsequent works, including DMQN [77], as effective alternatives to the standard Transformer. Notably, HSTU departs from conventional ULBM approaches by adopting a new sample organization scheme, strategically shifting from *multiple samples per user* to *a single sample per user*. This greatly reduces computational redundancy and enables autoregressive next-token prediction, allowing for more comprehensive modeling of interaction dependencies within user sequences. Following this direction, methods such as Fuxi family [107,108], GenRank [109] and MTGR [94] have demonstrated significant gains in industrial-scale recommender systems.

Scaled encoders represent the upper bound of effectiveness, but are typically applied to original behavior sequences whose lengths do not exceed ten thousand interactions. This reflects an inherent trade-off: modeling substantially longer sequences would incur prohibitive computational and inference costs. Nevertheless, by operating directly on complete lifelong behavior sequences, scaled encoders preserve richer information and therefore constitute a promising direction for future ULBM research.

4.1.4. Summary

From an EEB perspective, improving interaction modeling effectiveness in ULBM fundamentally amounts to enhancing modeling quality under fixed efficiency constraints, so that ROI can be sustained as user behavior sequences continue to grow. In practice, the widely adopted *preprocess-first, encode-later* paradigm achieves scalability by preprocessing ultra-long behavior sequences, but inevitably limits the granularity of interaction modeling due to constrained computation per token. Direct, single-stage modeling circumvents this issue by allocating substantially more computation to each token and enabling end-to-end exploitation of the original sequence. However, given the continuously growing length of user behavior histories and the stringent efficiency requirements of industrial-scale recommender systems, realizing more effective ULBM under practical constraints remains a substantial challenge. Nevertheless, this direction is widely regarded as a highly promising pathway toward advancing next-generation ULBM.

4.2. Fine-Grained User Interest Understanding

While the advancements discussed in Sec. 4.1, which progress from sum-pooling to Transformer architectures, have significantly expanded the *macro-modeling* capability of user behavior models, this remains far from sufficient. A high-capability model fed with coarse, entangled, or noisy data cannot fully realize its potential. To bridge the gap between model capability and data quality, recent research has shifted focus toward a more *micro-level* perspective: fine-grained deconstruction and analysis of user behavior sequences. This paradigm shift aims to disentangle the complexity of user intent by addressing four key challenges: modeling diverse behavioral types, decomposing multi-faceted interest subspaces, mitigating inherent noise within behavioral data, and proactively generating enhanced behavioral sequences. We organize the discussion around these four challenges, beginning with multi-behavior modeling.

 doi:10.20944/preprints202601.1559.v1

4.2.1. Multi-Behavior Modeling

Early work on user behavior sequence modeling often relied solely on click sequences. Subsequent studies incorporated a broader range of heterogeneous behaviors, such as like, add to cart, and purchase, yet these behaviors were typically encoded merely as discrete behavior types, which compresses these rich interaction signals into a homogeneous sequence and leads to the loss of critical information regarding intent intensity and preference semantics. Multi-behavior modeling aims not only to integrate heterogeneous signals to refine user representations but also to explicitly model the complex relationships between users, behaviors, and contextual information. The methodological evolution can be described as a progression: initially, each behavior type was modeled separately; later, methods began to capture dependencies among behaviors; subsequent approaches incorporated personalized behavior semantics to account for user-specific interpretation of actions; and the most recent methods explicitly model the scenario or intent under which a user performs a behavior. This evolution reflects a shift from coarse, homogeneous modeling toward precise, context-aware, and user-adaptive representations of multi-behavior sequences.

One approach to integrating multiple behaviors is through **graph-based** methods, which model the global, non-sequential correlations among different behavior types. This line of research constructs heterogeneous graphs to capture the global, non-sequential correlations among behaviors. By modeling the high-order connectivity between users and items across different behavior types, these methods can capture complex interaction patterns that linear sequences might miss. For instance, GNMR [47] and related graph-based works([48,110]) leverage Graph Neural Networks (GNNs) [111] to propagate embeddings across behavior-specific edges, effectively aggregating intent signals from the entire interaction network.

While global graph–based approaches capture high-level relational structures, they often overlook the detailed causal progression of user behaviors, and thus **cascading dependency** modeling focuses on the local behavioral sequence patterns (e.g., *Exposure* $\rightarrow$ *Click* $\rightarrow$ *Purchase*). Models such as CRGCN [112] and MB-CGCN [113] instantiate cascading dependencies by structuring message passing along the behavioral decision flow. CRGCN injects behavior-order priors through a sequence of residual blocks, but such stacking amplifies noise across stages. MB-CGCN replaces the residual stacking paradigm in CRGCN with a behavior-chaining design. It uses LightGCN [114] as the backbone to model each behavior type. The dependency between consecutive behaviors is captured by initializing the embedding of behavior $b+1$, denoted $\mathbf{e}_{(b+1,0)}$, with the learned representation of behavior b, denoted $\mathbf{e}_b$. Formally, taking the update of the user embedding as an example, this process can be expressed as:

$$\begin{aligned} \mathbf{e}^u_{(b)} &= \sum_{r=0}^{R} \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} \mathbf{e}^i_{(b,r)}, \\ \mathbf{e}^u_{(b+1,0)} &= \mathbf{W}^u_b \mathbf{e}^u_{(b)}, \end{aligned} \tag{21}$$

where N_u and N_i denotes the set of neighbors of user u and item i,respectively. R is the total number of layers, $\mathbf{e}^i_{(b,r)}$ is the representation of neighbor i at layer r for behavior b, and $\mathbf{W}^u_{(b)}$ is a learnable transformation matrix. The transformation can effectively distill useful features to facilitate the learning of the subsequent behavior's embedding, yet it inevitably introduces noise that may propagate through the sequence and degrade representation quality.

Another crucial insight in recent work, which we term **personalized behavior semantics**, is that the semantic implication of a specific behavior varies across users. For one user, "add-to-cart" might signal a definite purchase intent, while for another, it may merely serve as a bookmark. Approaches like MB-GMN [110], CML [115] utilize meta-learning or contrastive learning frameworks to learn personalized behavior semantics, ensuring that the model interprets the *meaning* of an action dynamically based on the user's specific context.

Along another dimension, a distinct line of work employs **special conditioning tokens** to guide the model in understanding and differentiating between the various intents within multi-behavior

 doi:10.20944/preprints202601.1559.v1

sequences. These tokens can represent either concrete behavioral scenarios (e.g., the `[stimulus_buy]` prompt in CMBR [116]) or abstract behavioral objectives (e.g., long-view and surprise moment in MP-Former [117]). This mechanism provides a unified framework for handling multi-behavior sequences and generating recommendations tailored to specific intents.

From an EEB perspective, most existing multi-behavior modeling approaches rely on graph-based structures. While these methods explicitly capture the relationships among users, behaviors, and contextual information, in ultra-long sequences at industrial scale, the massive number of neighbors and edges results in prohibitive graph construction and message propagation costs, as well as substantial memory usage and inference latency, rendering real-time deployment impractical. In contrast, conditioning token-based approaches offer a lightweight and adaptive mechanism, providing promising potential for guiding multi-behavior sequence modeling. Overall, these observations highlight that in ULBM, it is crucial to maximize the accuracy of user interest modeling while adhering to strict efficiency constraints, balancing the effectiveness of preference capture against computational and deployment costs.

4.2.2. Multi-Interest Modeling

While it helps clarify the *intensity* of intent, modeling multiple behaviors does not address the issue of *diversity*. A user's interests are often scattered and context-dependent. For instance, a user interested in basketball, coding, and classical music. Multi-interest modeling aims to capture both the diversity and evolution of user preferences. Since a user's interests are often scattered and context-dependent, representing them with a single dense vector may conflate distinct preferences. To address this, approaches can be conceptually divided into two complementary directions: **interest refinement**, which extracts and clarifies key aspects of historical behaviors, and **interest extrapolation**, which dynamically infers latent or evolving preferences beyond observed interactions.

A straightforward approach for handling long user histories is **key subsequence search**, which focuses on a critical subset of interactions under the assumption that interests strictly relevant to the current prediction are contained within a few essential behaviors. Methods in this category differ mainly in the type of search condition used to extract key subsequences from the original history. Approaches such as SIM [42] and TWIN [33] employ the candidate item as the query, whereas CoFARS [65] and LongRetriever [67] rely on contextual information. A more detailed discussion of these methods is provided in Sec. 3.1.1.

However, key subsequence search is limited by its reliance on a small subset of historical interactions and may fail to fully capture diverse user interests, motivating approaches that perform **comprehensive deconstruction** of the entire history. Clustering-based methods, such as MIND [17] and ComiRec [52], group user interactions into latent interest capsules using dynamic routing or soft clustering. Memory-based approaches, like MIMN [24], maintain external memory slots to track evolving preferences over time. Hierarchical decomposition models, including TWIN V2 [31], first compress lifelong user behaviors via offline hierarchical clustering to obtain interest clusters. It then employs a two-stage online search architecture over these clusters, effectively capturing interests at multiple temporal scales. Building on these structural paradigms, generative refinement methods have recently been proposed to further enhance the quality of multi-interest representations. For instance, DiffuMIN [118] treats aggregated interest embeddings as noisy latent factors and applies diffusion-based [119,120] denoising to produce high-fidelity virtual interest vectors, complementing traditional decomposition outputs.

Another line of work, which we refer to as **dynamic reasoning-based** modeling, argues that interest capture should not be treated as a static encoding but rather as a dynamic inference process. By performing implicit multi-step reasoning at inference time, these models iteratively refine their understanding of complex or long-tail preferences. For instance, the ReaRec [121] and OnePiece [122] frameworks autoregressively feed the encoder's output back into the model, employing iterative and progressive reasoning strategies to generate multi-view representations that capture subtle preference evolutions.

 doi:10.20944/preprints202601.1559.v1

From an EEB perspective, interest refinement methods achieve efficiency by either selecting key subsequences from long user histories or compressing behaviors into a moderate set of interest representations. This keeps online computation and memory overhead relatively low while still capturing the main aspects of user preferences. In contrast, interest extrapolation involves iterative, multi-step self-regressive updates of interest vectors, substantially increasing computational cost and inference latency, which poses challenges for industrial deployment. However, it provides more expressive and comprehensive multi-interest representations, enabling the model to capture subtle, long-tail, and evolving user preferences that are often missed by refinement-based methods, making it highly valuable despite the higher cost.

4.2.3. Behavior Sequence Denoising

Multi-interest modeling assumes that the observed sequence reflects the user's true intent. However, real-world data often contains noise such as accidental clicks, curiosity-driven views, or momentary drifts, which can distort interest decomposition and search. Denoising methods aim to purify the sequence by identifying and suppressing signals that do not correspond to the user's core preferences.

END4Rec [123] introduces a differentiable sequence labeling framework to dynamically assess and prune noisy behaviors. Similarly, recent works like SSDRec [124] and CDDRec [125] employ self-supervised signals or contrastive denoising to enhance the robustness of user representations against random interaction noise.

4.2.4. Behavior Sequence Generating

While denoising focuses on removing low-value signals and reducing data, another complementary strategy aims at constructing higher-value user sequences. For example, COFFEE [126] leverages additional data sources and richer semantic interactions to build more informative sequences, significantly enhancing ROI in real-world deployment. A more frontier direction involves active data augmentation via generative models. This paradigm shifts from discriminative filtering to generative synthesis, leveraging powerful distribution learners to reconstruct high-quality user representations.

In particular, some approaches use diffusion-based models [127,128], such as DiffuASR [129], to learn the distribution of user behavior embeddings or sequences and generate new, plausible sequences for data augmentation. Other methods, such as DR4SR [130], perform generative sequence synthesis via data distillation rather than diffusion, expanding the training data to improve the expressiveness, robustness, and generalization of downstream recommendation models.

4.2.5. Summary

The optimization of ULBM has shifted from a macroscopic focus on sequence encoding to a more microscopic, disentangled perspective. Key advancements now center around understanding and distinguishing the intensity of user intent through multi-behavior analysis, resolving conflicts between different preferences via interest decomposition, and improving the clarity of user signals by filtering out noise. Additionally, proactive techniques are being developed to generate more refined user representations. This fine-grained understanding forms the foundation for delivering precise and personalized recommendations. However, many existing approaches remain largely exploratory in nature. From an EEB perspective, due to their substantial computational cost and inference latency, they often struggle to meet the stringent efficiency constraints of large-scale industrial systems and have yet to demonstrate significant gains in real-world deployment. Nevertheless, these directions remain highly promising, offering valuable insights into the future evolution of effective and scalable ULBM.

4.3. Incorporating Worldwide Knowledge

Sec. 4.2 explored how to extract maximum utility from the user's *internal* behavioral data. Despite these advances, a fundamental limitation remains because the model's understanding is confined to the closed loop of historical interactions. Pure behavior-based models face a *semantic gap*: for

 doi:10.20944/preprints202601.1559.v1

instance, a model may learn that a user interacts with Nike Air Jordan and Slam Dunk, yet it does not capture the higher-level concept that both items belong to basketball culture. This lack of external context restricts generalization, reasoning, and cold-start capabilities. To overcome these limitations, recent research has focused on integrating *worldwide knowledge* into recommender systems. Among the diverse paradigms for incorporating worldwide knowledge, knowledge graphs constitute a canonical and extensively studied line of work.

4.3.1. E

arly efforts to bridge the semantic gap focused on *symbolic knowledge bases*. Knowledge graphs (KGs) [131,132] represent the external world using triples in the form of *head, relation, and tail*, providing reliable and verifiable semantic connections. By integrating knowledge graphs, models can propagate user preferences along semantic edges, such as from a movie to its director or genre.

Approaches such as KGCL [133] and KMCLR [134] exemplify this by performing contrastive learning jointly on the user-item interaction graph and the external knowledge graph. This injects semantic information into the latent space, alleviating data sparsity and providing path-based explainability.

4.3.2. Large Language Models

While knowledge graphs (KGs) provide structured knowledge bases, they are expensive to maintain and often suffer from limited coverage in open-domain scenarios. The advent of Large Language Models (LLMs) [135–137] has triggered a paradigm shift. Unlike KGs, LLMs serve as *parametric knowledge bases* that internalize vast amounts of open-world knowledge, enabling flexible semantic understanding and generation without relying on rigid schemas. Within ULBM, LLMs are particularly advantageous for modeling heterogeneous user behaviors, as they can capture semantic correlations across diverse types of interactions, temporal scales, and modalities.

The integration of LLMs into ULBM typically follows three progressive levels: augmentation, selection, and reasoning.

At the foundational level, LLMs act as **semantic encoders** that enrich traditional embeddings by bridging the gap between ID-based features and textual or multi-modal semantics. By encoding item descriptions or user profiles into high-quality semantic vectors, LLMs enhance the representation of both user and item features. For instance, EAGER [138] generates semantic representations from item texts to mitigate cold-start issues, while DMGIN [82] leverages multi-modal LLMs to compress ultra-long behavior sequences into semantically coherent interest groups, enabling the model to handle extensive histories with richer semantic awareness.

Building on this augmentation, LLMs can serve as **information purifiers**, addressing the challenges posed by long-term user behaviors. Traditional models may suffer from vanishing gradients, while LLMs face inherent context window limitations. In this role, LLMs act as intelligent retrievers or filters: by assessing the semantic relevance between a user's history and the current context, they curate a refined input for downstream models. For example, ReLLa [139] and ReLLaX [140] employ LLMs for zero-shot retrieval of the most relevant historical behaviors, and MISS [141] extends this to multi-modal scenarios, preserving key textual and visual cues within the Global Search Unit.

At the most advanced level, LLMs operate as **reasoning engines** capable of explicit cognitive inference. Here, LLMs go beyond feature extraction, interpreting user intent via structured deduction. By framing recommendation as a conditional generation task, these models leverage reasoning capabilities to infer preferences directly. KAR [142] elicits deep user preferences through prompting, converting reasoning chains into augmentation vectors. LIBER [97] incrementally reasons over long-term behavior sequences, dynamically tracking interest evolution. OneRec-Think [143] and RecZero [144] invoke logical inference to deduce latent user needs, marking a shift from pattern recognition toward cognitive understanding of heterogeneous behaviors.

 doi:10.20944/preprints202601.1559.v1

4.3.3. Summary

The incorporation of worldwide knowledge in ULBM has evolved from structured Knowledge Graphs to the unstructured, parametric knowledge of Large Language Models. LLMs are revolutionizing user lifelong behavior modeling by acting as semantic encoders, context-aware purifiers, and deep reasoning engines, enabling richer understanding of heterogeneous user behaviors across multiple modalities and temporal scales. From an EEB perspective, despite their expressive power, LLMs-based approaches incur substantial computational cost and latency, and may produce hallucinated outputs or suffer from limited controllability. These factors highlight an inherent trade-off: while LLMs enable deeper semantic and cross-behavior understanding, practical deployment requires consideration of efficiency and reliability. Overall, this trend marks a shift in recommender systems from purely statistical correlation matching toward systems capable of deep semantic interpretation and cognitive reasoning.

4.4. Discussion

ULBM has undergone advances in both its macro-level and micro-level modeling capabilities, enabling the construction of higher-quality user interest representations from original behavior sequences. Meanwhile, a growing body of work incorporates worldwide knowledge, allowing ULBM not only to exploit internal sequential signals but also to leverage external information for enhanced generalization and reasoning. With the rapid progress of LLMs, their powerful natural-language understanding capabilities have emerged as a new driving force for ULBM. In industrial-scale recommender systems, LLMs-integrated ULBM is increasingly able to absorb richer and more comprehensive information sources, pushing the field toward deeper semantic understanding and more forward-looking inference capabilities.

5. Dataset

Table 6. Summary of Public Datasets for User Lifelong Behavior Modeling.

Category	Dataset	#Users	#Items	#Interactions	Avg. Length	#Behavior Types	Applicability
General purpose	Amazon Books	694,897	686,623	10,053,086	14.47	5	○
	Movielens-1M	6,040	3,416	999,611	165.50	5	○
	Movielens-20M	138,493	26,744	20,000,263	144.41	5	○
E-commerce	Taobao	987,994	4,162,024	100,150,807	101.37	4	◐
	Tmall	424,170	1,090,390	54,925,331	129.49	4	◐
	Alibaba	1,141,729	461,527	700,000,000	613.11	4	●
Content Feeds	KuaiRand-Pure	27,285	7,551	1,436,609	52.65	7	◐
	KuaiRand-1K	1,000	4,369,953	11,713,045	11713.05	7	●
	KuaiRand-27K	27,285	32,038,725	322,278,385	11811.56	7	●
	KuaiRec-Small	1,411	3,327	4,676,570	3314.37	1	○
	KuaiRec-Big	7,176	10,728	12,530,806	1746.21	1	○
	KuaiSAR_S	25,877	3,026,189	5,059,169	195.51	4	◐
	KuaiSAR_R	25,877	4,046,367	14,605,716	564.43	4	●
	KuaiSAR_all	25,877	6,890,707	19,664,885	759.94	4	●
	TenRec-QK_video	5,022,750	3,753,436	493,458,970	98.24	4	○
	TenRec-QK_article	1,325,838	220,122	46,111,728	34.78	6	○
	TenRec-QB_video	34,240	130,637	2,442,299	71.33	4	○
	TenRec-QB_article	24,516	7,355	348,736	14.22	1	○
	Yambda_50M	10,000	934,057	48,000,000	4800.00	5	●
	Yambda_500M	100,000	3,004,578	480,000,000	4800.00	5	●
	Yambda_5B	1,000,000	9,390,623	4,800,000,000	4800.00	5	●

To systematically study user lifelong behavior modeling (ULBM), it is crucial to consider the characteristics of the datasets used for empirical evaluation. ULBM requires datasets that capture long-term user interactions and multi-behavior signals, which are essential for modeling evolving user interests and supporting both effectiveness and efficiency research.

 doi:10.20944/preprints202601.1559.v1

While industrial datasets provide rich, long-term interaction logs that naturally satisfy these requirements, they are typically proprietary and inaccessible to the research community. Consequently, publicly available datasets, spanning classical recommendation benchmarks, e-commerce platforms, and content feed logs, are often used as lower-bound proxies. These datasets vary substantially in scale, sequence length, interaction richness, and behavioral diversity, offering complementary perspectives for method validation, rapid prototyping, and controlled ablation studies.

In Table 6, we summarize the characteristics of these datasets and provide an applicability grading (●: Primary; ◐: Supplementary; ○: Not Advisable). It is important to note that most public datasets cover relatively short temporal horizons (typically one month), so the grading primarily reflects sequence length and behavioral richness, rather than true long-term engagement. We further report average sequence length as a coarse and comparable indicator of dataset scale. However, this statistic may not fully reflect the heavy-tailed distribution of user activity, particularly in content feeds datasets where a small fraction of users exhibit extremely long interaction histories. Despite these limitations, publicly available datasets remain valuable for reproducibility, low-barrier experimentation, and cross-study benchmarking, providing researchers with a practical entry point before deploying methods on industrial-scale logs.

5.1. General Purpose Datasets

At the early stage of empirical evaluation, general purpose datasets, such as Amazon Books [145], MovieLens-1M and MovieLens-20M [146], represent earlier yet still widely used benchmarks in recommendation research. Their interaction types are represented through explicit ratings, while the sequences of user interactions remain relatively short.

In typical ULBM pipelines, these ratings are converted into binary feedback by mapping low scores to negative and high scores to positive interactions. Notably, unlike most anonymized public benchmarks, the MovieLens series provides non-anonymous metadata, including movie titles and user-generated textual information such as reviews and tags. This property enables exploratory studies that incorporate external knowledge, for example by leveraging large language models to enrich item representations or contextualize user preferences in ULBM settings.

However, despite their widespread adoption in classical recommendation research, these datasets do not exhibit the key characteristics required for lifelong behavior modeling, such as long-term temporal continuity or heterogeneous multi-behavior signals. Consequently, they are considered *Not Advisable* for evaluating methods designed to capture lifelong user behavior. Nevertheless, due to their relatively small scale and well-controlled structure, they remain useful for preliminary ULBM experiments, sanity checks, and exploratory analyses.

5.2. E-Commerce Datasets

In contrast to general purpose datasets, large-scale e-commerce datasets provide richer interaction histories and longer temporal spans, making them more suitable for ULBM evaluation. Commercial platforms such as Taobao [147], Tmall [148], and Alibaba [149] provide large-scale logs capturing a variety of user interactions, including page views, cart additions, favorites, and purchases. These datasets share similar characteristics, but Alibaba stands out by providing longer user sequences, making it more suitable for ULBM.

Nevertheless, due to the fact that shopping activity tends to occur in bursts rather than as sustained daily engagement, even e-commerce datasets are limited in their ability to fully support ULBM. This distinction highlights that, although large-scale and multi-behavioral, e-commerce logs should be carefully evaluated when used for long-horizon, multi-behavior, or hierarchical modeling studies in ULBM frameworks.

 doi:10.20944/preprints202601.1559.v1

5.3. Content Feeds Datasets

Compared to general purpose and e-commerce datasets, content-oriented platforms such as video-sharing and digital reading services provide high-frequency, temporally continuous interaction logs, making them particularly suitable for studying long-term user behavior in ULBM.

Representative datasets include the KuaiRand [32], TenRec [150], and Yambda [151] series. Among them, the KuaiRand suite stands out as one of the most representative public benchmarks for ULBM. KuaiRand-1K and KuaiRand-27K contain ultra-long user behavior sequences, with average lengths exceeding ten thousand interactions, enabling the study of long-term preference dynamics. In contrast, the KuaiRand-Pure variant removes randomly recommended items, resulting in shorter but more curated sequences. An important characteristic of KuaiRand is its heterogeneous behavior signals, including six types of positive feedback and one negative feedback, which facilitates fine-grained modeling of user interests and their evolution over time.

The TenRec datasets cover two scenarios (QK and QB) across video-sharing and digital reading streams, and provide relatively rich heterogeneous feedback signals. However, user behavior sequences in TenRec are generally shorter, which limits their ability to capture long-term interest trajectories. As a result, these datasets are often used as complementary benchmarks rather than primary testbeds for ULBM.

The Yambda datasets, originating from a music streaming service, provide large-scale interaction logs with multiple feedback intensities and substantially longer user histories. Similar to KuaiRand, Yambda supports the analysis of long-horizon user behavior and evolving preferences.

Overall, content feeds datasets vary in their suitability for ULBM. Datasets that simultaneously provide ultra-long user sequences and heterogeneous behavior signals, such as KuaiRand and Yambda, are better aligned with the goal of understanding long-term user interest evolution. These properties are essential for evaluating ULBM methods, since capturing fine-grained behavioral dependencies and long-term preference dynamics lies at the core of the ULBM problem.

6. Future Work

6.1. Unified Cross Stage ULBM

Existing recommendation pipelines follow the retrieval-and-ranking paradigm, where user lifelong behavior modeling must be performed separately at each stage. As a result, user behavior histories are repeatedly processed, consuming substantial computational resources and reducing the Model FLOPs Utilization (MFU). A promising solution is to adopt a unified ULBM model offline, and then reuse the resulting user interest representations across multiple downstream tasks in different stages. This shift from *one model for one task* to *one model for multiple tasks* enables substantial savings in valuable computational resources and consequently improves MFU in industrial-scale recommender systems.

Only a small number of studies have begun exploring this direction [6], yet the paradigm of offline representation computation combined with online serving of downstream tasks offers clear advantages. It enables the use of substantially longer behavior sequences during offline modeling and supports more expressive architectures such as Transformers. However, existing methods fail to fully capture the interaction between the precomputed user representations and task-specific information, resulting in suboptimal representational quality for individual tasks. Inspired by the *pre-training and fine-tuning* paradigm [152,153], a promising avenue is to apply lightweight task-specific adaptation to the offline user representations within downstream tasks. Such fine-tuning can inject task-aware signals and more fully unlock the potential of unified ULBM models.

Furthermore, adopting a unified ULBM paradigm naturally stimulates a new wave of engineering optimizations. As user representations become reusable across stages, asynchronous interest representation refresh mechanisms can be introduced to update these representations without blocking online serving. Meanwhile, user representations can be stored, updated, and queried through a centralized representation service analogous to vector databases [154,155], enabling efficient access and cross-

 doi:10.20944/preprints202601.1559.v1

stage reuse. Together with multi-granularity caching, cross-stage memory sharing, and operator-level optimizations for ultra-long sequences, these advances significantly enhance the scalability of ULBM and support its deployment in large-scale industrial recommender systems.

6.2. LLMs Augmented ULBM

Large language models (LLMs) have recently made rapid progress and demonstrated strong potential across numerous domains [156–159]. While ULBM has long been deployed in industrial-scale recommender systems and proven its practical value, the integration of LLMs introduces new opportunities for capturing the dynamic evolution of user interests. Recent studies, such as CHIME [80] and DMGIN [82], take initial steps toward incorporating LLMs into ULBM by injecting multimodal knowledge and generating semantically enriched representations that enhance the expressive capacity of ULBM encoders.

Beyond producing enhanced representations, the reasoning capabilities of LLMs enable them to transcend pattern matching on historical interactions and construct structured interpretations of heterogeneous behaviors, contextual signals, and external knowledge. Beyond producing enhanced representations, the reasoning capabilities of LLMs enable ULBM methods to better understand user decision-making, facilitating more accurate and personalized recommendations by constructing structured interpretations of heterogeneous behaviors, contextual signals, and external knowledge.

An important extension of this direction involves employing LLMs as autonomous generative agents [160,161]. These agents can interact with users to directly acquire explicit preference information, which is more valuable than implicit feedback such as clicks or likes. However, such signals are often embedded in unstructured data, including text, images, and even emojis, requiring ULBM to transcend the conventional paradigm of learning solely from structured user logs and instead process unstructured information. This underscores the need to develop ULBM capable of handling multimodal data, performing long-text analysis, and potentially incorporating image recognition and sentiment analysis, thereby enabling the extraction of users' explicit preferences and substantially enhancing recommendation quality.

By combining semantic abstraction, reasoning enhancement, and dynamic interaction, this direction holds strong promise for constructing more adaptive, interpretable, and forward-looking ULBM frameworks in industrial-scale recommender systems.

6.3. Large-scale and End-to-End ULBM

As ULBM continues to advance, an important open question is whether scaling-law, regular performance gains that follow predictable statistical trends as data volume, behavior heterogeneity, and model capacity increase, may also emerge in this domain. Current developments suggest that ULBM is expanding along three principal axes: (I) modeling longer and more complete lifelong behavior sequences; (II) incorporating a broader spectrum of heterogeneous user actions; and (III) adopting larger-scale model architectures with greater representational capacity. These directions substantially increase the complexity and informational richness of ULBM, thereby creating conditions under which scaling-law could potentially be observed.

Such growing scale and complexity further amplify the limitations of the *preprocess first, encode later* paradigm, suggesting the need for end-to-end architectures that can directly leverage raw behavioral signals without incurring irreversible information loss. Admittedly, several recent studies have begun to explore end-to-end modeling for user lifelong behavior sequence, but the prohibitive computational cost has confined their sequence lengths to the scale of only a few thousand. Scaling end-to-end ULBM to much longer sequences while accommodating increasingly diverse user behavioral signals requires more computationally efficient architectures. Promising directions include replacing standard attention with linear-attention variants [162,163] or state space models [164,165], which enable user lifelong behavior modeling under linear-time complexity. In parallel, designing more efficient caching mechanisms is crucial for reusing offline computation during online inference, thereby achieving full decoupling between training and serving even under an end-to-end paradigm. Such advances

 doi:10.20944/preprints202601.1559.v1

would substantially reduce inference latency and make large-scale end-to-end user lifelong behavior modeling feasible in industrial-scale recommender systems.

6.4. Unified Cross Service ULBM

Contemporary internet companies increasingly operate multi-service ecosystems rather than isolated applications. These heterogeneous service verticals, such as video-sharing platforms, music streaming services, and digital reading applications, collectively shape a unified landscape of user engagement. The behavioral dependencies across services offer complementary perspectives on user interest formation and evolution, and incorporating such cross service interaction histories into ULBM would substantially enhance the fidelity and granularity of lifelong user representation. However, the expansion of multi-service ecosystems leads to a rapid proliferation of discrete ID features, which not only lack inherent semantic meaning but also impose substantial storage overhead. Moreover, the direct integration of cross-service data is constrained by privacy regulations, organizational boundaries, and naturally fragmented data silos.

A promising direction for mitigating the proliferation of discrete IDs is the adoption of semantic IDs. Semantic IDs employ a hierarchical codebook that maps semantically related entities to shared or partially shared codes. The resulting representation space naturally aligns with the coarse-to-fine structure of the item domain, enabling more efficient design and operation of the General Search Units and the Interest Reduction Units. Moreover, this mechanism facilitates unified modeling across modalities and service domains, allowing ULBM to effectively integrate heterogeneous data sources and construct a unified cross-service user representation.

Furthermore, federated learning provides a principled mechanism for privacy-preserving cross-service ULBM [166–168]. It enables collaborative model training across services without exposing raw user data, reconciling the need for ecosystem-level information integration with strict privacy constraints. This approach allows ULBM to learn aligned user representations across multi-service ecosystems while maintaining data privacy.

Advancing ULBM toward this privacy-preserving, semantically aligned, ecosystem-level paradigm holds strong promise for industrial-scale recommender systems, offering a path to break entrenched data silos, enable ecosystem-wide inference of long-term interests, and ultimately establish a unified cross-service feedback loop within large commercial platforms.

7. Future Work

7.1. Unified Cross Stage ULBM

Existing recommendation pipelines follow the retrieval-and-ranking paradigm, where user lifelong behavior modeling must be performed separately at each stage. As a result, user behavior histories are repeatedly processed, consuming substantial computational resources and reducing the Model FLOPs Utilization (MFU). A promising solution is to adopt a unified ULBM model offline, and then reuse the resulting user interest representations across multiple downstream tasks in different stages. This shift from *one model for one task* to *one model for multiple tasks* enables substantial savings in valuable computational resources and consequently improves MFU in industrial-scale recommender systems.

Only a small number of studies have begun exploring this direction [6], yet the paradigm of offline representation computation combined with online serving of downstream tasks offers clear advantages. It enables the use of substantially longer behavior sequences during offline modeling and supports more expressive architectures such as Transformers. However, existing methods fail to fully capture the interaction between the precomputed user representations and task-specific information, resulting in suboptimal representational quality for individual tasks. Inspired by the *pre-training and fine-tuning* paradigm [152,153], a promising avenue is to apply lightweight task-specific adaptation to the offline user representations within downstream tasks. Such fine-tuning can inject task-aware signals and more fully unlock the potential of unified ULBM models.

 doi:10.20944/preprints202601.1559.v1

Furthermore, adopting a unified ULBM paradigm naturally stimulates a new wave of engineering optimizations. As user representations become reusable across stages, asynchronous interest representation refresh mechanisms can be introduced to update these representations without blocking online serving. Meanwhile, user representations can be stored, updated, and queried through a centralized representation service analogous to vector databases [154,155], enabling efficient access and cross-stage reuse. Together with multi-granularity caching, cross-stage memory sharing, and operator-level optimizations for ultra-long sequences, these advances significantly enhance the scalability of ULBM and support its deployment in large-scale industrial recommender systems.

7.2. LLMs Augmented ULBM

Large language models (LLMs) have recently made rapid progress and demonstrated strong potential across numerous domains [156–159]. While ULBM has long been deployed in industrial-scale recommender systems and proven its practical value, the integration of LLMs introduces new opportunities for capturing the dynamic evolution of user interests. Recent studies, such as CHIME [80] and DMGIN [82], take initial steps toward incorporating LLMs into ULBM by injecting multimodal knowledge and generating semantically enriched representations that enhance the expressive capacity of ULBM encoders.

Beyond producing enhanced representations, the reasoning capabilities of LLMs enable them to transcend pattern matching on historical interactions and construct structured interpretations of heterogeneous behaviors, contextual signals, and external knowledge. Beyond producing enhanced representations, the reasoning capabilities of LLMs enable ULBM methods to better understand user decision-making, facilitating more accurate and personalized recommendations by constructing structured interpretations of heterogeneous behaviors, contextual signals, and external knowledge.

An important extension of this direction involves employing LLMs as autonomous generative agents [160,161]. These agents can interact with users to directly acquire explicit preference information, which is more valuable than implicit feedback such as clicks or likes. However, such signals are often embedded in unstructured data, including text, images, and even emojis, requiring ULBM to transcend the conventional paradigm of learning solely from structured user logs and instead process unstructured information. This underscores the need to develop ULBM capable of handling multimodal data, performing long-text analysis, and potentially incorporating image recognition and sentiment analysis, thereby enabling the extraction of users' explicit preferences and substantially enhancing recommendation quality.

By combining semantic abstraction, reasoning enhancement, and dynamic interaction, this direction holds strong promise for constructing more adaptive, interpretable, and forward-looking ULBM frameworks in industrial-scale recommender systems.

7.3. Large-scale and End-to-End ULBM

As ULBM continues to advance, an important open question is whether scaling-law, regular performance gains that follow predictable statistical trends as data volume, behavior heterogeneity, and model capacity increase, may also emerge in this domain. Current developments suggest that ULBM is expanding along three principal axes: (I) modeling longer and more complete lifelong behavior sequences; (II) incorporating a broader spectrum of heterogeneous user actions; and (III) adopting larger-scale model architectures with greater representational capacity. These directions substantially increase the complexity and informational richness of ULBM, thereby creating conditions under which scaling-law could potentially be observed.

Such growing scale and complexity further amplify the limitations of the *preprocess first, encode later* paradigm, suggesting the need for end-to-end architectures that can directly leverage raw behavioral signals without incurring irreversible information loss. Admittedly, several recent studies have begun to explore end-to-end modeling for user lifelong behavior sequence, but the prohibitive computational cost has confined their sequence lengths to the scale of only a few thousand. Scaling end-to-end ULBM to much longer sequences while accommodating increasingly diverse user behavioral signals

 doi:10.20944/preprints202601.1559.v1

requires more computationally efficient architectures. Promising directions include replacing standard attention with linear-attention variants [162,163] or state space models [164,165], which enable user lifelong behavior modeling under linear-time complexity. In parallel, designing more efficient caching mechanisms is crucial for reusing offline computation during online inference, thereby achieving full decoupling between training and serving even under an end-to-end paradigm. Such advances would substantially reduce inference latency and make large-scale end-to-end user lifelong behavior modeling feasible in industrial-scale recommender systems.

7.4. Unified Cross Service ULBM

Contemporary internet companies increasingly operate multi-service ecosystems rather than isolated applications. These heterogeneous service verticals, such as video-sharing platforms, music streaming services, and digital reading applications, collectively shape a unified landscape of user engagement. The behavioral dependencies across services offer complementary perspectives on user interest formation and evolution, and incorporating such cross service interaction histories into ULBM would substantially enhance the fidelity and granularity of lifelong user representation. However, the expansion of multi-service ecosystems leads to a rapid proliferation of discrete ID features, which not only lack inherent semantic meaning but also impose substantial storage overhead. Moreover, the direct integration of cross-service data is constrained by privacy regulations, organizational boundaries, and naturally fragmented data silos.

A promising direction for mitigating the proliferation of discrete IDs is the adoption of semantic IDs. Semantic IDs employ a hierarchical codebook that maps semantically related entities to shared or partially shared codes. The resulting representation space naturally aligns with the coarse-to-fine structure of the item domain, enabling more efficient design and operation of the General Search Units and the Interest Reduction Units. Moreover, this mechanism facilitates unified modeling across modalities and service domains, allowing ULBM to effectively integrate heterogeneous data sources and construct a unified cross-service user representation.

Furthermore, federated learning provides a principled mechanism for privacy-preserving cross-service ULBM [166–168]. It enables collaborative model training across services without exposing raw user data, reconciling the need for ecosystem-level information integration with strict privacy constraints. This approach allows ULBM to learn aligned user representations across multi-service ecosystems while maintaining data privacy.

Advancing ULBM toward this privacy-preserving, semantically aligned, ecosystem-level paradigm holds strong promise for industrial-scale recommender systems, offering a path to break entrenched data silos, enable ecosystem-wide inference of long-term interests, and ultimately establish a unified cross-service feedback loop within large commercial platforms.

8. Conclusions

In this work, we provide an industrial-perspective survey of user lifelong behavior modeling (ULBM), with a particular emphasis on the fundamental challenge of balancing efficiency and effectiveness. We summarize the core objectives and technical obstacles along these two dimensions, and conduct a systematic analysis of how representative methodologies address them, thereby improving the Return on Investment in industrial-scale recommender systems and driving the rapid evolution of this area. Reflecting on the trajectory of existing research, we outline promising future directions and observe with optimism the growing integration of ULBM with broader AI domains, which brings new theoretical insights, diversified applications, and cross-disciplinary opportunities. Grounded in the industrial viewpoint, we anticipate that ULBM will continue to adapt to evolving user behavior patterns and an increasingly complex recommendation ecosystem, ultimately advancing toward more robust, scalable, and intelligent lifelong modeling capabilities.

 doi:10.20944/preprints202601.1559.v1

References

1. Li, J.; Wang, M.; Li, J.; Fu, J.; Shen, X.; Shang, J.; McAuley, J. Text is all you need: Learning language representations for sequential recommendation. In Proceedings of the Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 1258–1267.
2. Yi, C.; Chen, D.; Guo, G.; Tang, J.; Wu, J.; Yu, J.; Zhang, M.; Dai, S.; Chen, W.; Yang, W.; et al. Recgpt technical report. *arXiv preprint arXiv:2507.22879* **2025**.
3. Zhang, Z.; Pei, H.; Guo, J.; Wang, T.; Feng, Y.; Sun, H.; Liu, S.; Sun, A. OneTrans: Unified Feature Interaction and Sequence Modeling with One Transformer in Industrial Recommender. *arXiv preprint arXiv:2510.26104* **2025**.
4. Deng, J.; Wang, S.; Cai, K.; Ren, L.; Hu, Q.; Ding, W.; Luo, Q.; Zhou, G. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* **2025**.
5. Huang, J.T.; Sharma, A.; Sun, S.; Xia, L.; Zhang, D.; Pronin, P.; Padmanabhan, J.; Ottaviano, G.; Yang, L. Embedding-based retrieval in facebook search. In Proceedings of the Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2553–2561.
6. Lyu, W.; Tyagi, D.; Yang, Y.; Li, Z.; Somani, A.; Shanmugasundaram, K.; Andrejevic, N.; Adeputra, F.; Zeng, C.; Singh, A.K.; et al. DV365: Extremely Long User History Modeling at Instagram. In Proceedings of the Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, 2025, pp. 4717–4727.
7. Guo, H.; Chen, B.; Tang, R.; Zhang, W.; Li, Z.; He, X. An embedding learning framework for numerical features in ctr prediction. In Proceedings of the Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2910–2918.
8. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the Proceedings of the 1st workshop on deep learning for recommender systems, 2016, pp. 7–10.
9. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* **2017**.
10. Huang, P.S.; He, X.; Gao, J.; Deng, L.; Acero, A.; Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 2333–2338.
11. Hambarde, K.A.; Proenca, H. Information retrieval: recent advances and beyond. *IEEE Access* **2023**, *11*, 76581–76604.
12. Grbovic, M.; Cheng, H. Real-time personalization using embeddings for search ranking at airbnb. In Proceedings of the Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 311–320.
13. Wang, K.; Wang, H.; Guo, W.; Liu, Y.; Lin, J.; Lian, D.; Chen, E. DLF: Enhancing explicit-implicit interaction via dynamic low-order-aware fusion for CTR prediction. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 2213–2223.
14. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.
15. Zhu, H.; Li, X.; Zhang, P.; Li, G.; He, J.; Li, H.; Gai, K. Learning tree-based deep model for recommender systems. In Proceedings of the Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1079–1088.
16. Yi, X.; Yang, J.; Hong, L.; Cheng, D.Z.; Heldt, L.; Kumthekar, A.; Zhao, Z.; Wei, L.; Chi, E. Sampling-bias-corrected neural modeling for large corpus item recommendations. In Proceedings of the Proceedings of the 13th ACM conference on recommender systems, 2019, pp. 269–277.
17. Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; Lee, D.L. Multi-interest network with dynamic routing for recommendation at Tmall. In Proceedings of the Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 2615–2623.
18. Huang, Y.; Cui, B.; Zhang, W.; Jiang, J.; Xu, Y. Tencentrec: Real-time stream recommendation in practice. In Proceedings of the Proceedings of the 2015 ACM SIGMOD international conference on management of data, 2015, pp. 227–238.

 doi:10.20944/preprints202601.1559.v1

19. Pal, A.; Eksombatchai, C.; Zhou, Y.; Zhao, B.; Rosenberg, C.; Leskovec, J. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In Proceedings of the Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 2311–2320.
20. Li, X.; Chen, B.; Guo, H.; Li, J.; Zhu, C.; Long, X.; Li, S.; Wang, Y.; Guo, W.; Mao, L.; et al. Inttower: the next generation of two-tower model for pre-ranking system. In Proceedings of the Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 3292–3301.
21. Qin, J.; Zhu, J.; Chen, B.; Liu, Z.; Liu, W.; Tang, R.; Zhang, R.; Yu, Y.; Zhang, W. Rankflow: Joint optimization of multi-stage cascade ranking systems as flows. In Proceedings of the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 814–824.
22. Zhou, R.; Wang, H.; Guo, W.; Jia, Q.; Xie, W.; Xu, X.; Liu, Y.; Lian, D.; Chen, E. MIT: A Multi-Tower Information Transfer Framework Based on Hierarchical Task Relationship Modeling. In Proceedings of the Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 651–660.
23. Liu, C.; Cao, J.; Huang, R.; Zheng, K.; Luo, Q.; Gai, K.; Zhou, G. KuaiFormer: Transformer-Based Retrieval at Kuaishou. *arXiv preprint arXiv:2411.10057* **2024**.
24. Pi, Q.; Bian, W.; Zhou, G.; Zhu, X.; Gai, K. Practice on long sequential user behavior modeling for click-through rate prediction. In Proceedings of the Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2671–2679.
25. He, Z.; Liu, W.; Guo, W.; Qin, J.; Zhang, Y.; Hu, Y.; Tang, R. A survey on user behavior modeling in recommender systems. *arXiv preprint arXiv:2302.11087* **2023**.
26. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE international conference on data mining (ICDM). IEEE, 2018, pp. 197–206.
27. Feng, Y.; Lv, F.; Shen, W.; Wang, M.; Sun, F.; Zhu, Y.; Yang, K. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* **2019**.
28. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* **2015**.
29. Liu, C.; Li, X.; Cai, G.; Dong, Z.; Zhu, H.; Shang, L. Noninvasive self-attention for side information fusion in sequential recommendation. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2021, Vol. 35, pp. 4249–4256.
30. Yuan, E.; Guo, W.; He, Z.; Guo, H.; Liu, C.; Tang, R. Multi-behavior sequential transformer recommender. In Proceedings of the Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, 2022, pp. 1642–1652.
31. Si, Z.; Guan, L.; Sun, Z.; Zang, X.; Lu, J.; Hui, Y.; Cao, X.; Yang, Z.; Zheng, Y.; Leng, D.; et al. Twin v2: Scaling ultra-long user behavior sequence modeling for enhanced ctr prediction at kuaishou. In Proceedings of the Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 4890–4897.
32. Gao, C.; Li, S.; Zhang, Y.; Chen, J.; Li, B.; Lei, W.; Jiang, P.; He, X. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In Proceedings of the Proceedings of the 31st ACM International Conference on Information and Knowledge Management, 2022, CIKM '22, p. 3953–3957. https://doi.org/10.1145/3511808.3557624.
33. Chang, J.; Zhang, C.; Fu, Z.; Zang, X.; Guan, L.; Lu, J.; Hui, Y.; Leng, D.; Niu, Y.; Song, Y.; et al. TWIN: TWo-stage interest network for lifelong user behavior modeling in CTR prediction at kuaishou. In Proceedings of the Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3785–3794.
34. Gupta, U.; Hsia, S.; Saraph, V.; Wang, X.; Reagen, B.; Wei, G.Y.; Lee, H.H.S.; Brooks, D.; Wu, C.J. Deeprecsys: A system for optimizing end-to-end at-scale neural recommendation inference. In Proceedings of the 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2020, pp. 982–995.
35. Naumov, M.; Mudigere, D.; Shi, H.J.M.; Huang, J.; Sundaraman, N.; Park, J.; Wang, X.; Gupta, U.; Wu, C.J.; Azzolini, A.G.; et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091* **2019**.
36. Liu, Z.; Zou, L.; Zou, X.; Wang, C.; Zhang, B.; Tang, D.; Zhu, B.; Zhu, Y.; Wu, P.; Wang, K.; et al. Monolith: real time recommendation system with collisionless embedding table. *arXiv preprint arXiv:2209.07663* **2022**.
37. Chen, Z.; Zhao, C.; Mo, K.C.; Jiang, Y.; Lee, J.H.; Chen, S.; Mahajan, K.C.; Jiang, N.; Ren, K.; Li, J.; et al. Massive Memorization with Hundreds of Trillions of Parameters for Sequential Transducer Generative Recommenders. *arXiv preprint arXiv:2510.22049* **2025**.

 doi:10.20944/preprints202601.1559.v1

38. Chen, X.; Li, Z.; Pan, W.; Ming, Z. A survey on multi-behavior sequential recommendation. *arXiv preprint arXiv:2308.15701* **2023**.
39. Pan, L.W.; Pan, W.K.; Wei, M.Y.; Yin, H.Z.; Ming, Z. A survey on sequential recommendation. *Frontiers of Computer Science* **2026**, *20*, 2003606.
40. Chen, S.; Xu, Z.; Pan, W.; Yang, Q.; Ming, Z. A survey on cross-domain sequential recommendation. *arXiv preprint arXiv:2401.04971* **2024**.
41. Kim, K.; Kim, S.; Lee, G.; Jung, J.; Shin, K. Multi-Behavior Recommender Systems: A Survey. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2025, pp. 435–452.
42. Pi, Q.; Zhou, G.; Zhang, Y.; Wang, Z.; Ren, L.; Fan, Y.; Zhu, X.; Gai, K. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In Proceedings of the Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2685–2692.
43. Zhou, G.; Deng, J.; Zhang, J.; Cai, K.; Ren, L.; Luo, Q.; Wang, Q.; Hu, Q.; Huang, R.; Wang, S.; et al. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* **2025**.
44. Covington, P.; Adams, J.; Sargin, E. Deep neural networks for youtube recommendations. In Proceedings of the Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 191–198.
45. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep interest network for click-through rate prediction. In Proceedings of the Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1059–1068.
46. Chen, Q.; Zhao, H.; Li, W.; Huang, P.; Ou, W. Behavior sequence transformer for e-commerce recommendation in alibaba. In Proceedings of the Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data, 2019, pp. 1–4.
47. Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Lu, M.; Bo, L. Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In Proceedings of the 2021 IEEE 37th international conference on data engineering (ICDE). IEEE, 2021, pp. 1931–1936.
48. Jin, B.; Gao, C.; He, X.; Jin, D.; Li, Y. Multi-behavior recommendation with graph convolutional networks. In Proceedings of the Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 659–668.
49. Wu, J.; Cai, R.; Wang, H. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In Proceedings of the Proceedings of The Web Conference 2020, 2020, pp. 2199–2209.
50. Ren, K.; Qin, J.; Fang, Y.; Zhang, W.; Zheng, L.; Bian, W.; Zhou, G.; Xu, J.; Yu, Y.; Zhu, X.; et al. Lifelong sequential modeling with personalized memorization for user response prediction. In Proceedings of the Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 565–574.
51. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
52. Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; Tang, J. Controllable multi-interest framework for recommendation. In Proceedings of the Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 2942–2951.
53. Qi, T.; Wu, F.; Wu, C.; Yang, P.; Yu, Y.; Xie, X.; Huang, Y. HieRec: Hierarchical user interest modeling for personalized news recommendation. *arXiv preprint arXiv:2106.04408* **2021**.
54. Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In Proceedings of the Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1161–1170.
55. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems* **2017**, *30*.
56. Jiang, Y.; Huang, C.; Huang, L. Adaptive graph contrastive learning for recommendation. In Proceedings of the Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, 2023, pp. 4252–4261.
57. Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* **2023**, *1*, 1–51.
58. Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; Li, Y. Sequential recommendation with graph neural networks. In Proceedings of the Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021, pp. 378–387.

 doi:10.20944/preprints202601.1559.v1

59. Hou, R.; Yang, Z.; Ming, Y.; Lu, H.; Zheng, Z.; Chen, Y.; Zeng, Q.; Chen, M. Cross-Domain LifeLong Sequential Modeling for Online Click-Through Rate Prediction. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5116–5125.
60. Zhu, Y.; Jiang, G.; Chen, J.; Zhang, F.; Wu, Q.; Liu, Z. Long-Term Interest Clock: Fine-Grained Time Perception in Streaming Recommendation System. In Proceedings of the Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 1554–1557.
61. Xia, X.; Joshi, S.; Rajesh, K.; Li, K.; Lu, Y.; Pancha, N.; Badani, D.; Xu, J.; Eksombatchai, P. TransAct V2: Lifelong User Action Sequence Modeling on Pinterest Recommendation. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 6881–6882.
62. Guo, T.; Li, X.; Yang, H.; Liang, X.; Yuan, Y.; Hou, J.; Ke, B.; Zhang, C.; He, J.; Zhang, S.; et al. Query-dominant User Interest Network for Large-Scale Search Ranking. In Proceedings of the Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 629–638.
63. Qin, J.; Zhang, W.; Wu, X.; Jin, J.; Fang, Y.; Yu, Y. User behavior retrieval for click-through rate prediction. In Proceedings of the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 2347–2356.
64. Xu, W.; Li, H.; Ou, B.; Xu, L.; Qin, Y.; Su, R.; Xu, R. GIST: Cross-Domain Click-Through Rate Prediction via Guided Content-Behavior Distillation. *arXiv preprint arXiv:2507.05142* **2025**.
65. Feng, Z.; Xie, J.; Li, K.; Qin, Y.; Wang, P.; Li, Q.; Yin, B.; Li, X.; Lin, W.; Wang, S. Context-based Fast Recommendation Strategy for Long User Behavior Sequence in Meituan Waimai. In Proceedings of the Companion Proceedings of the ACM Web Conference 2024, 2024, pp. 355–363.
66. Cai, Y.; Hou, J.; Zhu, Y.; Nie, Y. Interest Changes: Considering User Interest Life Cycle in Recommendation System. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 2592–2596.
67. Ren, Q.; Chai, Z.; Xiao, X.; Zheng, Y.; Wu, D. LongRetriever: Towards Ultra-Long Sequence based Candidate Retrieval for Recommendation. *arXiv preprint arXiv:2508.15486* **2025**.
68. Chen, Q.; Pei, C.; Lv, S.; Li, C.; Ge, J.; Ou, W. End-to-end user behavior retrieval in click-through rateprediction model. *arXiv preprint arXiv:2108.04468* **2021**.
69. Cao, Y.; Zhou, X.; Feng, J.; Huang, P.; Xiao, Y.; Chen, D.; Chen, S. Sampling is all you need on modeling long-term user behaviors for CTR prediction. In Proceedings of the Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 2974–2983.
70. Xu, X.; Wang, H.; Guo, W.; Zhang, L.; Yang, W.; Yu, R.; Liu, Y.; Lian, D.; Chen, E. Multi-granularity interest retrieval and refinement network for long-term user behavior modeling in ctr prediction. In Proceedings of the Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1, 2025, pp. 2745–2755.
71. Feng, N.; Pan, J.; Wu, J.; Chen, B.; Wang, X.; Li, Q.; Hu, X.; Jiang, J.; Long, M. Long-Sequence Recommendation Models Need Decoupled Embeddings. *arXiv preprint arXiv:2410.02604* **2024**.
72. Lv, X.; Cao, J.; Guan, S.; Zhou, X.; Qi, Z.; Zang, Y.; Li, M.; Wang, B.; Gai, K.; Zhou, G. MARM: Unlocking the Future of Recommendation Systems through Memory Augmentation and Scalable Complexity. *arXiv preprint arXiv:2411.09425* **2024**.
73. Yan, J.; Jiang, L.; Cui, J.; Zhao, Z.; Bin, X.; Zhang, F.; Liu, Z. Trinity: Syncretizing Multi-/Long-Tail/Long-Term Interests All in One. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6095–6104.
74. Meng, Y.; Guo, C.; Hu, X.; Deng, H.; Cao, Y.; Liu, T.; Zheng, B. User Long-Term Multi-Interest Retrieval Model for Recommendation. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 1112–1116.
75. Ju, C.M.; Neves, L.; Kumar, B.; Collins, L.; Zhao, T.; Qiu, Y.; Dou, Q.; Zhou, Y.; Nizam, S.; Ozturk, R.A.; et al. Learning universal user representations leveraging cross-domain user intent at snapchat. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 4345–4349.
76. Zhou, W.J.; Zheng, Y.; Feng, Y.; Ye, Y.; Xiao, R.; Chen, L.; Yang, X.; Xiao, J. ENCODE: Breaking the Trade-Off Between Performance and Efficiency in Long-Term User Behavior Modeling. *IEEE Transactions on Knowledge and Data Engineering* **2024**.
77. Wei, Z.; Liu, Q.; Xie, Q. Deep Multiple Quantization Network on Long Behavior Sequence for Click-Through Rate Prediction. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 3090–3094.

 doi:10.20944/preprints202601.1559.v1

78. Guo, T.; Yang, Z.; Zeng, Q.; Chen, M. Context-Aware Lifelong Sequential Modeling for Online Click-Through Rate Prediction. *arXiv preprint arXiv:2502.12634* **2025**.
79. Li, K.; Tang, Y.; Cheng, Y.; Bai, Y.; Zeng, Y.; Wang, C.; Liu, X.; Jiang, P. VQL: An End-to-End Context-Aware Vector Quantization Attention for Ultra-Long User Behavior Modeling. *arXiv preprint arXiv:2508.17125* **2025**.
80. Bai, Y.; Xiang, R.; Li, K.; Tang, Y.; Cheng, Y.; Liu, X.; Jiang, P.; Gai, K. Chime: A compressive framework for holistic interest modeling. *arXiv preprint arXiv:2504.06780* **2025**.
81. Xu, S.; Wang, S.; Guo, D.; Guo, X.; Xiao, Q.; Li, F.; Luo, C. An Efficient Large Recommendation Model: Towards a Resource-Optimal Scaling Law. *arXiv preprint arXiv:2502.09888* **2025**.
82. Wei, Z.; Xie, Q.; Liu, Q. DMGIN: How Multimodal LLMs Enhance Large Recommendation Models for Lifelong User Post-click Behaviors. *arXiv preprint arXiv:2508.21801* **2025**.
83. Liu, Q.; Hou, X.; Jin, H.; Chen, X.; Chen, J.; Lian, D.; Wang, Z.; Cheng, J.; Lei, J. Deep Group Interest Modeling of Full Lifelong User Behaviors for CTR Prediction. *arXiv preprint arXiv:2311.10764* **2023**.
84. Xia, Y.; Zhong, R.; Gu, H.; Yang, W.; Lu, C.; Jiang, P.; Gai, K. Hierarchical tree search-based user lifelong behavior modeling on large language model. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 1758–1767.
85. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International conference on machine learning. PmLR, 2021, pp. 8748–8763.
86. Dai, Z.; Lai, G.; Yang, Y.; Le, Q. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Advances in neural information processing systems* **2020**, *33*, 4271–4282.
87. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data* **2021**, *8*, 53.
88. Li, X.; Liang, J.; Liu, X.; Zhang, Y. Adversarial filtering modeling on long-term user behavior sequences for click-through rate prediction. In Proceedings of the Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1969–1973.
89. Pan, M.; Yang, X.; Qiao, N.; Wang, D.; Mei, F.; Zhao, X.; Xu, S. Hierarchical User Long-term Behavior Modeling for Click-Through Rate Prediction. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 2880–2884.
90. Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems* **2022**, *35*, 16344–16359.
91. Zhai, J.; Liao, L.; Liu, X.; Wang, Y.; Li, R.; Cao, X.; Gao, L.; Gong, Z.; Gu, F.; He, M.; et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152* **2024**.
92. Rabe, M.N.; Staats, C. Self-attention does not need $O(n^2)$ memory. *arXiv preprint arXiv:2112.05682* **2021**.
93. Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691* **2023**.
94. Han, R.; Yin, B.; Chen, S.; Jiang, H.; Jiang, F.; Li, X.; Ma, C.; Huang, M.; Li, X.; Jing, C.; et al. Mtgr: Industrial-scale generative recommendation framework in meituan. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 5731–5738.
95. Chai, Z.; Ren, Q.; Xiao, X.; Yang, H.; Han, B.; Zhang, S.; Chen, D.; Lu, H.; Zhao, W.; Yu, L.; et al. Longer: Scaling up long sequence modeling in industrial recommenders. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 247–256.
96. Chen, X.; Rajesh, K.; Lawhon, M.; Wang, Z.; Li, H.; Li, H.; Joshi, S.V.; Eksombatchai, P.; Yang, J.; Hsu, Y.P.; et al. Pinfm: foundation model for user activity sequences at a billion-scale visual discovery platform. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 381–390.
97. Zhu, C.; Quan, S.; Chen, B.; Lin, J.; Cai, X.; Zhu, H.; Li, X.; Xi, Y.; Zhang, W.; Tang, R. LIBER: Lifelong User Behavior Modeling Based on Large Language Models. *arXiv preprint arXiv:2411.14713* **2024**.
98. Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Heek, J.; Xiao, K.; Agrawal, S.; Dean, J. Efficiently scaling transformer inference. *Proceedings of machine learning and systems* **2023**, *5*, 606–624.
99. Song, X.; Li, X.; Hu, J.; Wen, H.; Chen, Z.; Zhang, Y.; Zeng, X.; Zhang, J. Lrea: Low-rank efficient attention on modeling long-term user behaviors for ctr prediction. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 2843–2847.

 doi:10.20944/preprints202601.1559.v1

100. Jun, H.; Cho, J.; Lee, K.; Son, H.Y.; Kim, K.; Jin, H.; Kim, K. Hbm (high bandwidth memory) dram technology and architecture. In Proceedings of the 2017 IEEE International Memory Workshop (IMW). IEEE, 2017, pp. 1–4.
101. Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; Gai, K. Deep interest evolution network for click-through rate prediction. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, Vol. 33, pp. 5941–5948.
102. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv 2014. *arXiv preprint arXiv:1412.3555* **2014**, *1412*.
103. Wu, B.; Yang, F.; Chan, Z.; Gu, Y.R.; Feng, J.; Yi, C.; Sheng, X.R.; Zhu, H.; Xu, J.; Ye, M.; et al. MUSE: A Simple Yet Effective Multimodal Search-Based Framework for Lifelong User Interest Modeling. *arXiv preprint arXiv:2512.07216* **2025**.
104. Huang, T.; Zhang, Z.; Zhang, J. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In Proceedings of the Proceedings of the 13th ACM conference on recommender systems, 2019, pp. 169–177.
105. Guan, L.; Yang, J.Q.; Zhao, Z.; Zhang, B.; Sun, B.; Luo, X.; Ni, J.; Li, X.; Qi, Y.; Fan, Z.; et al. Make It Long, Keep It Fast: End-to-End 10k-Sequence Modeling at Billion Scale on Douyin. *arXiv preprint arXiv:2511.06077* **2025**.
106. Zhu, J.; Fan, Z.; Zhu, X.; Jiang, Y.; Wang, H.; Han, X.; Ding, H.; Wang, X.; Zhao, W.; Gong, Z.; et al. Rankmixer: Scaling up ranking models in industrial recommenders. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 6309–6316.
107. Ye, Y.; Guo, W.; Chin, J.Y.; Wang, H.; Zhu, H.; Lin, X.; Ye, Y.; Liu, Y.; Tang, R.; Lian, D.; et al. FuXi-α: Scaling Recommendation Model with Feature Interaction Enhanced Transformer. In Proceedings of the Companion Proceedings of the ACM on Web Conference 2025, 2025, pp. 557–566.
108. Ye, Y.; Guo, W.; Wang, H.; Zhu, H.; Ye, Y.; Liu, Y.; Guo, H.; Tang, R.; Lian, D.; Chen, E. Fuxi-\beta: Towards a lightweight and fast large-scale generative recommendation model. *arXiv preprint arXiv:2508.10615* **2025**.
109. Huang, Y.; Chen, Y.; Cao, X.; Yang, R.; Qi, M.; Zhu, Y.; Han, Q.; Liu, Y.; Liu, Z.; Yao, X.; et al. Towards Large-scale Generative Ranking. *arXiv preprint arXiv:2505.04180* **2025**.
110. Xia, L.; Xu, Y.; Huang, C.; Dai, P.; Bo, L. Graph meta network for multi-behavior recommendation. In Proceedings of the Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021, pp. 757–766.
111. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* **2020**, *32*, 4–24.
112. Yan, M.; Cheng, Z.; Gao, C.; Sun, J.; Liu, F.; Sun, F.; Li, H. Cascading residual graph convolutional network for multi-behavior recommendation. *ACM Transactions on Information Systems* **2023**, *42*, 1–26.
113. Cheng, Z.; Han, S.; Liu, F.; Zhu, L.; Gao, Z.; Peng, Y. Multi-behavior recommendation with cascading graph convolution networks. In Proceedings of the Proceedings of the ACM Web Conference 2023, 2023, pp. 1181–1189.
114. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 639–648.
115. Wei, W.; Huang, C.; Xia, L.; Xu, Y.; Zhao, J.; Yin, D. Contrastive meta learning with behavior multiplicity for recommendation. In Proceedings of the Proceedings of the fifteenth ACM international conference on web search and data mining, 2022, pp. 1120–1128.
116. Gou, Y.; Yao, Y.; Zhang, Z.; Wu, Y.; Hu, Y.; Zhuang, F.; Liu, J.; Xu, Y. Controllable multi-behavior recommendation for in-game skins with large sequential model. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 4986–4996.
117. Sun, Y.; Huang, S.; Che, L.; Lu, H.; Luo, Q.; Gai, K.; Zhou, G. MPFormer: Adaptive Framework for Industrial Multi-Task Personalized Sequential Retriever. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 2832–2841.
118. Lai, W.; Jin, B.; Zhang, Y.; Zheng, Y.; Zhao, R.; Dong, J.; Lei, J.; Wang, X. Modeling Long-term User Behaviors with Diffusion-driven Multi-interest Network for CTR Prediction. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 289–298.
119. Li, Z.; Sun, A.; Li, C. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems* **2023**, *42*, 1–28.

 doi:10.20944/preprints202601.1559.v1

120. Xie, W.; Wang, H.; Zhang, L.; Zhou, R.; Lian, D.; Chen, E. Breaking determinism: Fuzzy modeling of sequential recommendation using discrete state space diffusion model. *Advances in Neural Information Processing Systems* **2024**, *37*, 22720–22744.
121. Tang, J.; Dai, S.; Shi, T.; Xu, J.; Chen, X.; Chen, W.; Jian, W.; Jiang, Y. Think Before Recommend: Unleashing the Latent Reasoning Power for Sequential Recommendation. CoRR abs/2503.22675 (2025). doi: 10. 48550. *arXiv preprint ARXIV.2503.22675* **2025**.
122. Dai, S.; Tang, J.; Wu, J.; Wang, K.; Zhu, Y.; Chen, B.; Hong, B.; Zhao, Y.; Fu, C.; Wu, K.; et al. OnePiece: Bringing Context Engineering and Reasoning to Industrial Cascade Ranking System. *arXiv preprint arXiv:2509.18091* **2025**.
123. Han, Y.; Wang, H.; Wang, K.; Wu, L.; Li, Z.; Guo, W.; Liu, Y.; Lian, D.; Chen, E. End4rec: Efficient noise-decoupling for multi-behavior sequential recommendation. *arXiv preprint arXiv:2403.17603* **2024**.
124. Zhang, C.; Han, Q.; Chen, R.; Zhao, X.; Tang, P.; Song, H. Ssdrec: Self-augmented sequence denoising for sequential recommendation. In Proceedings of the 2024 IEEE 40th International Conference on Data Engineering (ICDE). IEEE, 2024, pp. 803–815.
125. Wang, Y.; Liu, Z.; Yang, L.; Yu, P.S. Conditional denoising diffusion for sequential recommendation. In Proceedings of the Pacific-Asia conference on knowledge discovery and data mining. Springer, 2024, pp. 156–169.
126. Roychowdhury, S.; Wang, D.; Ge, Q.; Mu, J.; Reddy, S. COFFEE: COdesign Framework for Feature Enriched Embeddings in Ads-Ranking Systems. *arXiv preprint arXiv:2601.02807* **2026**.
127. Wang, W.; Xu, Y.; Feng, F.; Lin, X.; He, X.; Chua, T.S. Diffusion recommender model. In Proceedings of the Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval, 2023, pp. 832–841.
128. Xie, W.; Zhou, R.; Wang, H.; Shen, T.; Chen, E. Bridging user dynamics: Transforming sequential recommendations with schrödinger bridge and diffusion models. In Proceedings of the Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 2618–2628.
129. Liu, Q.; Yan, F.; Zhao, X.; Du, Z.; Guo, H.; Tang, R.; Tian, F. Diffusion augmentation for sequential recommendation. In Proceedings of the Proceedings of the 32nd ACM International conference on information and knowledge management, 2023, pp. 1576–1586.
130. Yin, M.; Wang, H.; Guo, W.; Liu, Y.; Zhang, S.; Zhao, S.; Lian, D.; Chen, E. Dataset regeneration for sequential recommendation. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 3954–3965.
131. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Computing Surveys (Csur)* **2021**, *54*, 1–37.
132. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 950–958.
133. Yang, Y.; Huang, C.; Xia, L.; Li, C. Knowledge graph contrastive learning for recommendation. In Proceedings of the Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, 2022, pp. 1434–1443.
134. Xuan, H.; Liu, Y.; Li, B.; Yin, H. Knowledge enhancement for contrastive multi-behavior recommendation. In Proceedings of the Proceedings of the sixteenth ACM international conference on web search and data mining, 2023, pp. 195–203.
135. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
136. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* **2023**.
137. Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. A survey on large language models for recommendation. *World Wide Web* **2024**, *27*, 60.
138. Wang, Y.; Xun, J.; Hong, M.; Zhu, J.; Jin, T.; Lin, W.; Li, H.; Li, L.; Xia, Y.; Zhao, Z.; et al. Eager: Two-stream generative recommender with behavior-semantic collaboration. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 3245–3254.

 doi:10.20944/preprints202601.1559.v1

139. Lin, J.; Shan, R.; Zhu, C.; Du, K.; Chen, B.; Quan, S.; Tang, R.; Yu, Y.; Zhang, W. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In Proceedings of the Proceedings of the ACM Web Conference 2024, 2024, pp. 3497–3508.
140. Shan, R.; Zhu, J.; Lin, J.; Zhu, C.; Chen, B.; Tang, R.; Yu, Y.; Zhang, W. Full-Stack Optimized Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. *ACM Transactions on Recommender Systems* **2025**.
141. Guo, C.; She, J.; Cai, K.; Wang, S.; Hu, Q.; Luo, Q.; Zhou, G.; Gai, K. MISS: Multi-Modal Tree Indexing and Searching with Lifelong Sequential Behavior for Retrieval Recommendation. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 5683–5690.
142. Xi, Y.; Liu, W.; Lin, J.; Cai, X.; Zhu, H.; Zhu, J.; Chen, B.; Tang, R.; Zhang, W.; Yu, Y. Towards open-world recommendation with knowledge augmentation from large language models. In Proceedings of the Proceedings of the 18th ACM Conference on Recommender Systems, 2024, pp. 12–22.
143. Liu, Z.; Wang, S.; Wang, X.; Zhang, R.; Deng, J.; Bao, H.; Zhang, J.; Li, W.; Zheng, P.; Wu, X.; et al. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639* **2025**.
144. Kong, X.; Jiang, J.; Liu, B.; Xu, Z.; Zhu, H.; Xu, J.; Zheng, B.; Wu, J.; Wang, X. Think before Recommendation: Autonomous Reasoning-enhanced Recommender. *arXiv preprint arXiv:2510.23077* **2025**.
145. McAuley, J.; Targett, C.; Shi, Q.; Van Den Hengel, A. Image-based recommendations on styles and substitutes. In Proceedings of the Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015, pp. 43–52.
146. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* **2015**, *5*, 1–19.
147. Tianchi. User Behavior Data from Taobao for Recommendation, 2018.
148. Tianchi. IJCAI-15 Repeat Buyers Prediction Dataset, 2018.
149. Tianchi. Taobao Display Advertising Click-Through Rate Prediction Dataset, 2018.
150. Yuan, G.; Yuan, F.; Li, Y.; Kong, B.; Li, S.; Chen, L.; Yang, M.; YU, C.; Hu, B.; Li, Z.; et al. Tenrec: A Large-scale Multipurpose Benchmark Dataset for Recommender Systems. In Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.
151. Ploshkin, A.; Tytskiy, V.; Pismenny, A.; Baikalov, V.; Taychinov, E.; Permiakov, A.; Burlakov, D.; Krofto, E. Yambda-5B—A Large-Scale Multi-Modal Dataset for Ranking and Retrieval. In Proceedings of the Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 894–901.
152. Qiu, Z.; Wu, X.; Gao, J.; Fan, W. U-BERT: Pre-training user representations for improved recommendation. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021, Vol. 35, pp. 4320–4327.
153. Wang, J.; Zeng, Z.; Wang, Y.; Wang, Y.; Lu, X.; Li, T.; Yuan, J.; Zhang, R.; Zheng, H.T.; Xia, S.T. Missrec: Pre-training and transferring multi-modal interest-aware sequence representation for recommendation. In Proceedings of the Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 6548–6557.
154. Ma, L.; Zhang, R.; Han, Y.; Yu, S.; Wang, Z.; Ning, Z.; Zhang, J.; Xu, P.; Li, P.; Ju, W.; et al. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *arXiv preprint arXiv:2310.11703* **2023**.
155. Pan, J.J.; Wang, J.; Li, G. Survey of vector database management systems. *The VLDB Journal* **2024**, *33*, 1591–1615.
156. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* **2023**.
157. Thirunavukarasu, A.J.; Ting, D.S.J.; Elangovan, K.; Gutierrez, L.; Tan, T.F.; Ting, D.S.W. Large language models in medicine. *Nature medicine* **2023**, *29*, 1930–1940.
158. Demszky, D.; Yang, D.; Yeager, D.S.; Bryan, C.J.; Clapper, M.; Chandhok, S.; Eichstaedt, J.C.; Hecht, C.; Jamieson, J.; Johnson, M.; et al. Using large language models in psychology. *Nature Reviews Psychology* **2023**, *2*, 688–701.
159. Zhao, Z.; Fan, W.; Li, J.; Liu, Y.; Mei, X.; Wang, Y.; Wen, Z.; Wang, F.; Zhao, X.; Tang, J.; et al. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering* **2024**, *36*, 6889–6907.
160. Vullam, N.; Vellela, S.S.; Reddy, V.; Rao, M.V.; SK, K.B.; et al. Multi-agent personalized recommendation system in e-commerce based on user. In Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE, 2023, pp. 1194–1199.

 doi:10.20944/preprints202601.1559.v1

161. Wang, Y.; Jiang, Z.; Chen, Z.; Yang, F.; Zhou, Y.; Cho, E.; Fan, X.; Lu, Y.; Huang, X.; Yang, Y. Recmind: Large language model powered agent for recommendation. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 4351–4364.
162. Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; Li, H. Efficient attention: Attention with linear complexities. In Proceedings of the Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 3531–3539.
163. Team, K.; Zhang, Y.; Lin, Z.; Yao, X.; Hu, J.; Meng, F.; Liu, C.; Men, X.; Yang, S.; Li, Z.; et al. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692* **2025**.
164. Gu, A.; Goel, K.; Ré, C. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396* **2021**.
165. Gu, A.; Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In Proceedings of the First conference on language modeling, 2024.
166. Chronis, C.; Varlamis, I.; Himeur, Y.; Sayed, A.N.; Al-Hasan, T.M.; Nhlabatsi, A.; Bensaali, F.; Dimitrakopoulos, G. A survey on the use of federated learning in privacy-preserving recommender systems. *IEEE Open Journal of the Computer Society* **2024**, *5*, 227–247.
167. Reddy, M.S.; Karnati, H.; Sundari, L.M. Transformer based federated learning models for recommendation systems. *IEEE Access* **2024**.
168. Feng, C.; Feng, D.; Huang, G.; Liu, Z.; Wang, Z.; Xia, X.G. Robust privacy-preserving recommendation systems driven by multimodal federated learning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**, *36*, 8896–8910.