RESEARCH-ARTICLE

# Transformers are Good Clusterers for Lifelong User Behavior Sequence Modeling

**XINGMEI WANG**, University of Science and Technology of China, Hefei, Anhui, China

**SHIYAO WANG**, Kuaishou, Beijing, China

**WUCHAO LI**, University of Science and Technology of China, Hefei, Anhui, China

**JIAXIN DENG**, Kuaishou, Beijing, China

**SONG LU**, Kuaishou, Beijing, China

**DEFU LIAN**, University of Science and Technology of China, Hefei, Anhui, China

**View all**



**Published:** 10 November 2025

**Citation in BibTeX format**

CIKM '25: The 34th ACM International Conference on Information and Knowledge Management
*November 10 - 14, 2025*
*Seoul, Republic of Korea*

**Conference Sponsors:**
SIGWEB
SIGIR

# Transformers are Good Clusterers for Lifelong User Behavior Sequence Modeling

Xingmei Wang
University of Science and Technology of China
Hefei, China
xingmeiwang@mail.ustc.edu.cn

Shiyao Wang
Kuaishou Technology
Beijing, China
wangshiyao08@kuaishou.com

Wuchao Li
University of Science and Technology of China
Hefei, China
liwuchao@mail.ustc.edu.cn

Jiaxin Deng
Kuaishou Technology
Beijing, China
dengjiaxin03@kuaishou.com

Song Lu
Kuaishou Technology
Beijing, China
lusong@kuaishou.com

Defu Lian*
University of Science and Technology of China
Hefei, China
liandefu@ustc.edu.cn

Guorui Zhou
Kuaishou Technology
Beijing, China
zhouguorui@kuaishou.com

## Abstract

Modeling user long-term behavior sequences is critical for enhancing Click-Through Rate (CTR) prediction. Existing methods typically employ two cascaded search units—General Search Unit (GSU) for rapid retrieval and Exact Search Unit (ESU) for precise modeling—to balance efficiency and effectiveness. However, they are constrained to recent behaviors due to computational limitations. Clustering user behaviors offers a potential solution, enabling GSU to access lifelong behaviors while maintaining inference efficiency, but current clustering approaches often lack generalizability, or fail to remain effective in high-dimensional data due to non-end-to-end clustering and recommendation. Given that centroids in clustering group similar data points based on proximity, similar to how queries function in transformers, we can integrate the learning of queries with CTR tasks in an end-to-end manner, shifting clustering from meaningless Euclidean distances to meaningful semantic distances. Therefore, we propose **C-Former**, a transformer-based clustering model specifically designed for modeling lifelong behavior sequences. The ***C-Former encoder*** leverages a group of learnable clustering anchor points that access the lifelong user behaviors to extract personalized interests. Then, the ***C-Former decoder*** reconstructs lifelong user behaviors based on the compact output of the encoder. The reconstruction and orthogonal loss ensure that centroids are informative and diverse in capturing user preferences. Clustering is further guided by supervisory signals from CTR, establishing an end-to-end framework. The proposed C-Former achieves linear time complexity in training with respect to sequence length and significantly reduces inference latency by directly utilizing cached centroids. Experiments on four benchmark datasets demonstrate the effectiveness of C-Former for lifelong user behavior sequence modeling. The code is available at https://github.com/pepsi2222/C-Former.

## CCS Concepts

• **Information systems** → **Recommender systems**; **Clustering and classification**.

## Keywords

Click-Through Rate Prediction, Lifelong User Behavior, Clustering, Recommender System

**ACM Reference Format:**
Xingmei Wang, Shiyao Wang, Wuchao Li, Jiaxin Deng, Song Lu, Defu Lian, and Guorui Zhou. 2025. Transformers are Good Clusterers for Lifelong User Behavior Sequence Modeling. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761294

## 1 Introduction

Recommender systems (RS) facilitate the delivery of personalized content to users by systematically analyzing their browsing history and demographic data, thereby presenting items that are in accordance with their individual preferences [18, 45]. In this framework, Click-Through Rate (CTR) prediction is of paramount importance, as it evaluates the likelihood of user engagement with a specific item. User behavior sequence modeling significantly enhances CTR prediction by capturing users' evolving interests, enabling richer

*Corresponding author.


*CIKM '25, Seoul, Republic of Korea.*

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761294

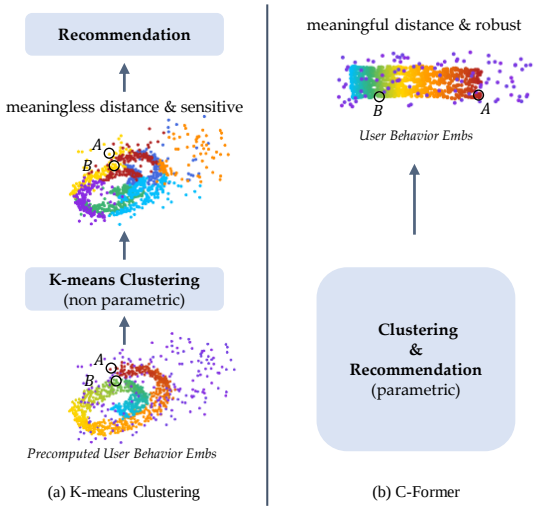



**Figure 1: K-means vs. C-Former. K-means provides a non-parametric approach that clusters precomputed representations for recommendation, while C-Former integrates representation learning and deep clustering end-to-end.**

profiling and greater recommendation accuracy. However, while longer sequences provide more contextual information and improve predictions, they also increase computational costs and inference latency. Therefore, it is crucial to balance the benefits of sequence length with computational efficiency during inference.

Existing long-sequence modeling typically involves two cascaded search units: 1) General Search Unit (GSU), which broadly searches raw sequences to rapidly retrieve the top-$K$ relevant behaviors. 2) Exact Search Unit (ESU), which precisely models the relationship between the candidate item and the retrieved behaviors, often using a targeted attention network. The cascade design is essential due to the high computational cost of ESU, making its direct application to lifelong sequences infeasible. While GSU reduces the sequence size, enabling feasible exact searches [4, 5, 27], it remains limited by computational overhead. This often restricts searches to recent behaviors, preventing access to entire lifelong behaviors. A solution is to cluster lifelong behaviors and provide centroids to GSU, maintaining input length while capturing full sequence information, such as DGIN [22] and TWINv2 [33]. However, DGIN groups behaviors by item IDs, making it suitable only for scenarios with repeated interactions. TWINv2, on the other hand, employs non-parametric K-means to cluster precomputed recommendation representations, but its distance measured by Euclidean metric loses meaning in high-dimensional spaces, particularly due to TWINv2's non-end-to-end approach to representation learning and clustering. For instance, as illustrated in Figure 1, two points (A and B) that are spatially proximate but semantically dissimilar are incorrectly grouped into the same cluster. Additionally, K-means is sensitive to noise and outliers, further limiting its effectiveness.

In clustering, centroids interact with data points, grouping similar ones based on their proximity to the corresponding centroids. This concept inspires us about the cross-attention mechanism in transformers, where queries act like centroids to interact with keys to compute attention scores and determine how much focus should be placed on each value. Moreover, the learning of queries, akin to centroids, can be seamlessly integrated end-to-end with downstream tasks like CTR prediction, effectively shifting clustering from meaningless Euclidean to meaningful semantic distance. While some works [25, 38] have also attempted to combine transformers for clustering, they require category-labeled datasets or assume equal data division, making them impractical and inapplicable for lifelong behavior modeling and recommendation tasks. Therefore, we propose a general transformer-based clustering model, termed **C-Former**, which is designed to model lifelong user behavior sequences and derive high-quality centroids through clustering. To be specific, **firstly**, we propose a *C-Former encoder* based on Transformers, where a set of clustering anchor points as learnable queries interact with user behaviors as keys and values, automatically aggregating relevant user behaviors and extracting underlying interests from behavior sequences; **secondly**, we design a *C-Former decoder* that retains the Transformer architecture, reconstructing lifelong user behaviors based on these clustering anchor points output from the encoder; **thirdly**, We implement a denoising mechanism that assigns behaviors to clusters, and by aggregating intra-cluster behaviors, it produces denoised interest centroids. **Finally**, for training the model, we incorporate a reconstruction loss to enforce that interest centroids accurately reconstruct lifelong user behaviors and an orthogonal loss to promote diversity in user preferences. To ensure an end-to-end framework, clustering is further guided by supervisory signals from RS. The training time complexity of C-Former is linear in relation to sequence length and the number of clusters, which ensures efficient training. During inference, the use of cached centroids makes latency independent of sequence length, thereby ensuring availability for serving.

The main contributions of this paper are summarized as follows:

- We propose C-Former, a general clustering framework for modeling lifelong user behavior sequences. It provides a comprehensive representation of user preferences by spanning lifelong behavior sequences while maintaining linear training complexity with respect to sequence length and enabling low-latency inference through the use of centroids.
- We have carefully crafted reconstruction and orthogonal losses to improve the quality of centroids, ensuring they are both informative for reconstructing lifelong user behavior sequences and diverse in capturing user preferences.
- To validate the effectiveness of C-Former, experiments are conducted on four benchmark datasets. The results demonstrate that C-Former effectively clusters lifelong user behaviors and significantly enhances performance for CTR prediction.

## 2 Related Works

### 2.1 Click-Through-Rate Prediction

Click-through rate (CTR) prediction [15, 24, 39, 40, 43, 44] aims to estimate the probability of a user engaging with a specific item by clicking on it. Over time, CTR models have evolved significantly, with foundational architectures such as Factorization Machines (FM) [31] introducing second-order feature interactions. WideDeep [7] and DeepFM [14] extended this by incorporating

deep learning components to capture high-level feature interactions, while PNN [29] enhanced feature interaction by coupling first-order and second-order features. AutoInt [34] leveraged self-attention mechanisms to dynamically weight feature interactions. DCN [39] and DCNv2 [40] automated the discovery of meaningful feature crossings.

The focus of research has shifted toward modeling user behavior sequences. DIN [48] initiated this trend by using attention mechanisms to emphasize candidate-relevant behaviors. DIEN [47] advanced this approach with a two-layer GRU to model temporal shifts and extract deeper user interests. DSIN [13] further enhanced sequence modeling by segmenting behavior sequences into sessions, combining self-attention and Bi-LSTM for session representations, and using target attention to derive session-level interests. BST [6] enhanced this paradigm by leveraging transformer architectures to capture complex sequential patterns.

## 2.2 Long-Term User Behavior Modeling

Recent advancements in CTR prediction have extended beyond short-term user behavior modeling to incorporate long-term behavior sequences, capturing extended temporal patterns and richer user interest representations. MIMN [26] introduced a memory-based framework for long-term behavior modeling, where updates are triggered by real-time user interactions, eliminating delays during inference. Subsequent approaches adopted a two-stage GSU-ESU retrieval strategy, differing primarily in the first GSU stage retrieval mechanism. UBR4CTR [28] adaptively learns query features and employs BM25 for behavior selection. SIM Hard [27] uses category-based hard retrieval, while SIM Soft [27] retrieves top-k behaviors based on the inner product between candidate items and historical behaviors. ETA [5] utilizes hashing and Hamming distance for efficient retrieval, while SDIM [3] selects behaviors through multiple hash collisions and aggregates their embeddings. TWIN [4] proposed a twin-attention mechanism to unify the two stages. MIRRN [42] employs multi-interest retrieval and refinement networks to extract diverse interests from long sequences. DGIN [22] groups behaviors using relevance keys to filter redundant items, and TWINv2 [33] extended TWIN [4] by using hierarchical k-means clustering to retrieve from longer behavior sequences.

However, most existing approaches [3–5, 26–28, 42] are constrained by computational overhead, often limiting searches to recent behaviors. Clustering methods for lifelong behaviors either lack generalizability across diverse scenarios [22] or rely on non-parametric K-means to cluster precomputed recommendation representations [33], which results in meaningless distance metrics in high-dimensional spaces and sensitive to noise and outliers.

# 3 Preliminary

## 3.1 Problem Statement

Consider the task of clustering behaviors within a behavior sequence $S = \{b_1, b_2, \ldots, b_L\}$ for each user, we aim to partition these behaviors into $J$ clusters. Here, $b_l$ represents the $l$-th behavior in the sequence, and $L$ is the total length of the sequence. Each cluster is characterized by a centroid, denoted as $\boldsymbol{\mu}_j$ for $j = 1, 2, \ldots, J$.

The objective is to determine an optimal partitioning of behaviors into $J$ clusters such that the centroids $\{\boldsymbol{\mu}_j\}$ effectively represent the entire behavior sequence $S$, thereby optimizing performance for downstream tasks, such as the CTR prediction task.

## 3.2 Transformer

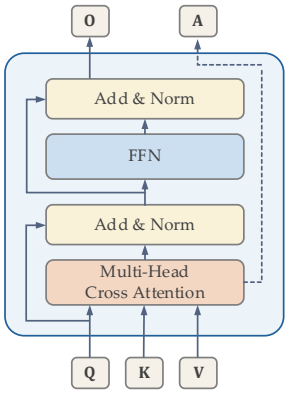



**Figure 2: The architecture of a single-layer Transformer.**

The Transformer, introduced by [37], has become a cornerstone in natural language processing (NLP) [10, 30] and RS [9, 12, 17, 20, 35]. In this work, we design an adaptation, as depicted in Figure 2. This version enables the query to selectively focus on different segments of the key sequence, while improving efficiency by excluding self-attention.

At the heart of the decoder lies the multi-head cross-attention mechanism, which enables one sequence to attend to another. This mechanism computes attention scores between two distinct input sequences, typically a query sequence $\mathbf{Q} \in \mathbb{R}^{L_q \times d_{\text{model}}}$ and a key-value sequence $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{L_k \times d_{\text{model}}}$. The query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ are projected into $h$ separate subspaces of dimensions $d_k$, $d_k$, and $d_v$, respectively. For each head, the attention function is computed in parallel as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

$$\mathbf{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{2}$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, \mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, \mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are projection matrices. The outputs from all heads are concatenated and then linearly projected to produce the final representation:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O \tag{3}$$

For brevity, we denote the whole module involving this multi-head cross attention mechanism and other parts like *Feed-Forward Network (FFN)* and *Add & Norm* illustrated in Figure 2 as:

$$\mathbf{O}, \mathbf{A} = \text{Transformer}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \tag{4}$$

where $\mathbf{O}$ is the output representation and $\mathbf{A} = \text{softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})$ is the attention score.

In a Transformer, queries interact with keys to compute attention scores, determining the focus on each value. This process is analogous to centroids in clustering, where centroids interact with data

points to group similar ones based on their proximity. These similar concepts inspire the use of Transformers to naturally facilitate clustering.

## 4 Methodology

In this section, we present C-Former, a novel clustering method designed to model lifelong user behavior sequences. This approach leverages the Transformer architecture to model lifelong user behaviors for end-to-end clustering and recommendation.

### 4.1 C-Former

We first introduce the architecture of C-Former, as illustrated in Figure 3. It adopts an encoder-decoder framework: the encoder captures distinct user interests through a set of learnable clustering anchor points, and the decoder reconstructs lifelong user behaviors with these clustering anchor points. Detailed descriptions of them are provided in Section 4.1.1 and 4.1.2, respectively.

*4.1.1 Encoder.* Leveraging the Transformer's ability to aggregate semantically similar elements, we design the C-Former encoder as a multi-layer Transformer to facilitate clustering. As illustrated in the lower left corner of Figure 3, **the encoder employs user behavior sequence embeddings $\mathbf{E}^S$ as keys and values, while a set of learnable clustering anchor points $\mathbf{Q}_i$ serve as queries for the $i$-th layer.** This configuration enables each clustering anchor point to automatically identify and aggregate similar user behaviors based on attention scores. The output of the $i$-th Transformer layer $\mathbf{O}_i$, derived from aggregated similar behaviors, becomes the refined query $\mathbf{Q}_{i+1}$ for the $(i+1)$-th layer, progressively uncovering underlying themes or interests. Through iterative refinement, the clustering anchor points are anchored to distinct clusters that capture user interests. The encoding process is formalized as:

$$\begin{aligned} \mathbf{O}_i, \mathbf{A}_i &= \text{Transformer}(\mathbf{Q}_i, \mathbf{E}^S, \mathbf{E}^S), \\ \mathbf{Q}_{i+1} &= \mathbf{O}_i, \text{ for } i = 1, 2, \ldots, M \end{aligned} \tag{5}$$

where $\mathbf{Q}_i \in \mathbb{R}^{J \times d}$ is the input query for the $i$-th layer, while $\mathbf{E}^S \in \mathbb{R}^{L \times d}$ serves as both the key and value inputs. The initial query $\mathbf{Q}_1$ is randomly initialized. The output of the $i$-th layer includes the aggregated representation $\mathbf{O}_i \in \mathbb{R}^{J \times d}$ and attention weights $\mathbf{A}_i \in \mathbb{R}^{J \times L}$. Here, $d$ is the embedding dimension, and $M$ is the number of encoder layers.

The final output of the C-Former encoder consists of the refined clustering anchor points $\mathbf{O}^{\text{enc}}$ and the attention weights $\mathbf{A}$ generated by the last Transformer layer:

$$\mathbf{O}^{\text{enc}} = \mathbf{O}_M \tag{6}$$

$$\mathbf{A} = \mathbf{A}_M \tag{7}$$

*4.1.2 Decoder.* To ensure that the refined clustering anchor points preserve maximal information about the lifelong user behavior sequence, we introduce a decoder that reconstructs the sequence based on these anchor points. The decoder retains the Transformer architecture, as illustrated in the green box in Figure 3. **Unlike the encoder, it treats user behaviors as queries and the refined clustering anchor points $\mathbf{O}^{\text{enc}}$ as both keys and values.** This approach allows each user behavior to be reconstructed by combining the refined clustering anchor points with varying attention weights. The decoding process is formalized as:

$$\tilde{\mathbf{Q}}_1 = \mathbf{E}^S \tag{8}$$

$$\begin{aligned} \tilde{\mathbf{O}}_i, \tilde{\mathbf{A}}_i &= \text{Transformer}(\tilde{\mathbf{Q}}_i, \mathbf{O}^{\text{enc}}, \mathbf{O}^{\text{enc}}), \\ \tilde{\mathbf{Q}}_{i+1} &= \tilde{\mathbf{O}}_i, \text{ for } i = 1, 2, \ldots, N \end{aligned} \tag{9}$$

$$\mathbf{O}^{\text{dec}} = \tilde{\mathbf{O}}_N \tag{10}$$

where $\tilde{\mathbf{Q}}_i \in \mathbb{R}^{L \times d}$ represents the input query for the $i$-th layer, while the refined clustering anchor points $\mathbf{O}^{\text{enc}}$ serve as both the key and value inputs. The output of the $i$-th layer includes the reconstructed behaviors $\tilde{\mathbf{O}}_i \in \mathbb{R}^{L \times d}$ and the attention weights $\tilde{\mathbf{A}}_i \in \mathbb{R}^{L \times J}$. The decoder comprises $N$ layers.

*4.1.3 Denoise & Assignment.* To achieve a more precise cluster representation, we denoise the refined clustering anchor points. Noise arises because each anchor point is a weighted sum of user behaviors—where relevant behaviors are assigned higher weights and irrelevant ones receive minimal weights. Consequently, each anchor point also incorporates contributions from irrelevant behaviors, introducing noise. To address this, we explicitly assign each behavior $b_i$ to its corresponding cluster $c_i$ based on the $\mathbf{A}$, thereby identifying and aggregating relevant behaviors while eliminating trivial contributions. The assignment and aggregation process is formalized as:

$$c_i = \text{argmax } \mathbf{A}_{:,i} \tag{11}$$

$$\boldsymbol{\mu}_j = \sum_{i=1}^{L} \mathbf{A}_{j,i} \mathbf{E}_i^S \, \mathbb{I}(c_i = j) \tag{12}$$

where $\mathbf{A}_{:,i} \in \mathbb{R}^J$ is the $i$-th column of $\mathbf{A}$, $\boldsymbol{\mu}_j$ denotes the centroid of the $j$-th cluster, $\mathbf{E}_i^S$ denotes the embedding of the $i$-th behavior $b_i$ and $\mathbb{I}(\cdot)$ is the indicator function.

Since each behavior may serve multiple purposes, it can be assigned to more than one cluster. In such cases, Eq. (11) and (12) can be adjusted as:

$$\boldsymbol{c}_i = \text{Top-K}(\mathbf{A}_{:,i}, k) \tag{13}$$

$$\boldsymbol{\mu}_j = \sum_{i=1}^{L} \mathbf{A}_{j,i} \mathbf{E}_i^S \, \mathbb{I}(j \in \boldsymbol{c}_i) \tag{14}$$

where Top-K$(\cdot, k)$ retrieves the indices of the top-$k$ values.

### 4.2 End-to-End Clustering & Recommendation

We now present the training objective for the C-Former model, which comprises three distinct loss components: a reconstruction loss to preserve the information of the user behavior sequence, an orthogonal loss to promote diversity, and a binary cross-entropy loss to facilitate end-to-end clustering and recommendation.

*4.2.1 Reconstruction Loss.* To ensure that the reconstructed sequence $O^{\text{dec}}$, generated by the centroids, retains as much information as possible from the original user behavior sequence $E^S$, we introduce a reconstruction loss between $E^S$ and $O^{\text{dec}}$. This loss encourages the centroids to be informative:

$$\mathcal{L}_{recon} = \|\mathbf{E}^S - \mathbf{O}^{\text{dec}}\|^2 \tag{15}$$

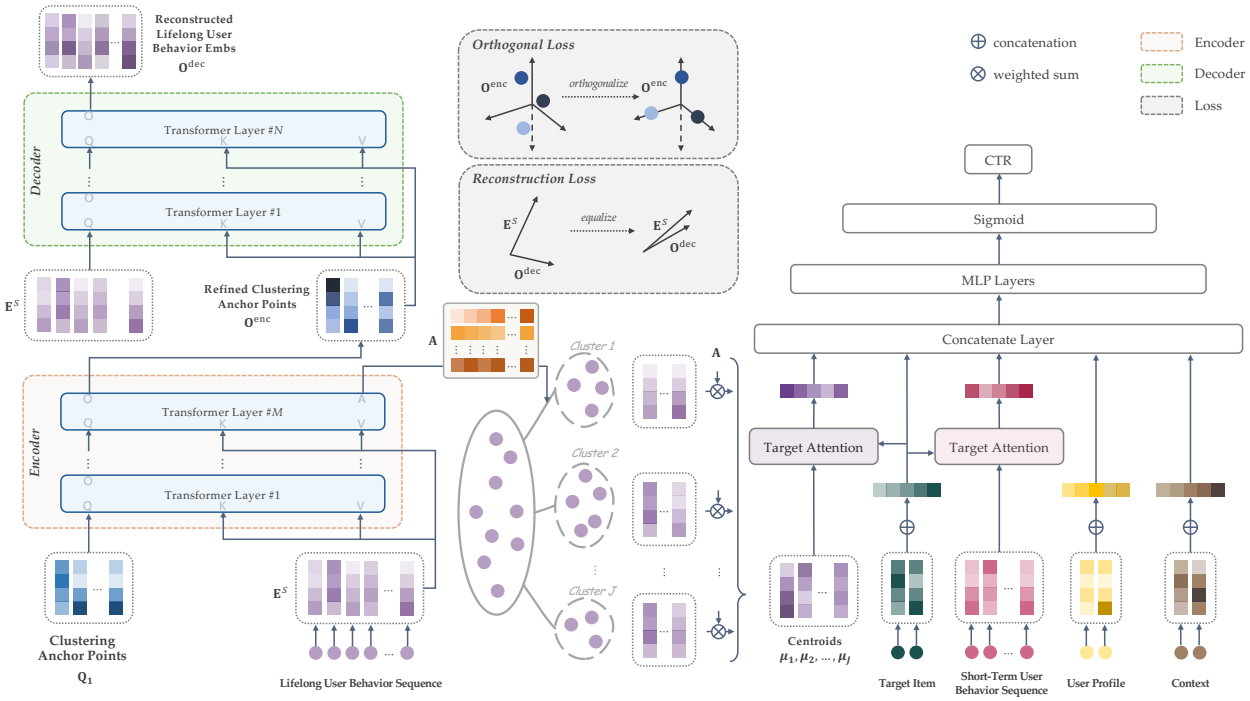



**Figure 3: The overall framework of C-Former. C-Former comprises four main components: an encoder with $M$ Transformer layers, a decoder with $N$ Transformer layers, a denoise & assignment mechanism, and optimization losses. The encoder captures user interests by leveraging a set of learnable clustering anchor points $Q_i$ as queries and sequence embeddings $E^S$ as keys and values. The decoder reconstructs user behaviors based on the refined clustering anchor points $O^{enc}$, with user behaviors as queries and $O^{enc}$ as keys and values. The attention weights A assign each behavior to its corresponding cluster, and the weighted aggregation of intra-cluster behaviors generates the final centroids for the CTR task.**

*4.2.2 Orthogonal Loss.* To promote diverse user interests, the clustering anchor points should be maximally separated. To achieve this, we introduce an orthogonal loss applied to the refined clustering anchor points, enforcing orthogonality between them:

$$\mathcal{L}_{orth} = \left\| \frac{\mathbf{O}^{\text{enc}}(\mathbf{O}^{\text{enc}})^T}{\|\mathbf{O}^{\text{enc}}\|_2^2} - \mathbf{I} \right\|_F \tag{16}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, defined as the square root of the sum of squares of all matrix elements, and $\mathbf{I}$ denotes the identity matrix.

*4.2.3 Binary Cross-Entropy Loss.* To facilitate end-to-end clustering and recommendation, we incorporate a supervised binary cross-entropy loss between the ground truth label and the predicted probability of user engagement with a target item:

$$\mathcal{L}_{bce} = -\frac{1}{N} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} [y \log p(\boldsymbol{x}) + (1-y)\log(1-p(\boldsymbol{x}))] \tag{17}$$

where $\mathcal{T}$ is the training set of size $N$, $\boldsymbol{x}$ represents the model input, comprising the target item, user profile, context, short-term behaviors, and lifelong behaviors, $y \in \{0,1\}$ is the ground truth label, and $p(\boldsymbol{x})$ is the predicted CTR, formalized as:

$$p(\boldsymbol{x}) = \sigma\left(\text{MLP}\left(\mathbf{E}^t; \mathbf{E}^u; \mathbf{E}^c; \mathbf{E}^{\text{short}}; \mathbf{E}^{\text{lifelong}} \mid \boldsymbol{x}\right)\right) \tag{18}$$

where $\mathbf{E}^t$, $\mathbf{E}^u$, and $\mathbf{E}^c$ denote embeddings of the target item, the user profile, and the context, respectively. $\mathbf{E}^{\text{short}}$ represents short-term user interests, derived through target attention applied to the short-term user behavior sequence, while $\mathbf{E}^{\text{lifelong}}$ represents lifelong user interests, derived by applying target attention to the centroids.

The composite loss function of C-Former is:

$$\mathcal{L} = \mathcal{L}_{recon} + \alpha\mathcal{L}_{orth} + \beta\mathcal{L}_{bce} \tag{19}$$

## 4.3 Complexity Analysis

We present a comprehensive theoretical analysis of inference time complexity across different methods in Table 1. The transformer demonstrates linear time complexity with respect to query and key lengths, resulting in C-Former's centroid computation complexity of $O(BJLd)$, which scales linearly with both the original sequence length $L$ and the number of centroids $J$. Notably, the centroid calculation is independent of target items and solely relies on the lifelong user behavior sequence, which exhibits significantly lower refresh rates compared to short-term behaviors. As the framework's short-term behavior sequence modeling has already facilitated real-time interaction processing, the centroids of lifelong behavior sequences need not be computed in real-time during inference. Consequently,

**Table 1: Inference time complexity of different methods. $B$ is the batch size, $d$ is the model's hidden size, $L$ and $K$ are the lengths of the original and retrieved user behavior sequences, respectively (where $K$ is also equal to the number of centroids $J$), $A$ is the size of the attribute inverted index, $m$ is the number of hash functions, $c$ is the number of cross features, and $n$ is the size of the clusters. For methods other than DIN, we exclude their short-term sequence modeling overhead here, as it is minor.**

| Model | Inference Time Complexity |
|---|---|
| DIN | $O(BLd)$ |
| SIM Hard | $O(B\log(A) + BKd)$ |
| SIM Soft | $O(BLd + BKd)$ |
| ETA | $O(BLm + BKd)$ |
| SDIM | $O(Bm\log(d))$ |
| MIRRN | $O(BLm + BK\log(K)d + BKd^2)$ |
| TWIN | $O(BL + BcLd + BKd)$ |
| TWINv2 | $O(\frac{BL}{n} + \frac{BcLd}{n} + BKd)$ |
| C-Former | $O(BKd)$ |

C-Former's inference time complexity reduces to $O(BJd)$. When setting $J$ equal to $K$, C-Former's inference time complexity is $O(BKd)$, which is sufficiently low for practical applicability.

## 5 Experiments

In this section, we conduct extensive experiments to assess C-Former by answering these questions.

- **RQ1:** How does C-Former perform against the state-of-the-art user behavior modeling models for RS?
- **RQ2:** How do different components influence C-Former?
- **RQ3:** How does C-Former's inference efficiency?
- **RQ4:** How does C-Former perform at different sequence lengths?
- **RQ5:** What is the impact of different hyperparameter settings on C-Former performance?
- **RQ6:** Does C-Former demonstrate superior clustering performance compared to other clustering methods for user behaviors?

### 5.1 Experimental Settings

*5.1.1 Datasets.* We conduct extensive experiments on three datasets, with statistical details provided in Table 2.

**Taobao** [1] comprises activities of about 1 million users on a large e-commerce platform between November 25, 2017, and December 3, 2017, and is widely utilized as a benchmark [3, 5, 22, 26, 42].

**Alipay** [2] gathers users' online payment transactions conducted on Alipay over a five-month period.

**Tmall** [3] contains shopping records of users on the Tmall e-commerce platform from May 11, 2015, to November 11, 2015.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[2]https://tianchi.aliyun.com/dataset/dataDetail?dataId=53
[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42

**Table 2: Statistics of datasets.**

| Dataset | #Users | #Items | #Interaction | Avg. Seq. Length | Max. Seq. Length |
|---|---|---|---|---|---|
| Taobao | 987,994 | 4,162,024 | 100,150,807 | 101 | 827 |
| Alipay | 498,308 | 2,200,291 | 35,179,371 | 70 | 83,293 |
| Tmall | 424,179 | 1,090,390 | 54,925,331 | 129 | 13,713 |
| XLong | 20,000 | 3,263,616 | 20,000,000 | 1,000 | 1,000 |

**XLong** [4] comprises user click logs sampled from a large e-commerce platform during April-September 2018, with each user having approximately 1,000 interaction sequences.

*5.1.2 Baselines.* Our method is benchmarked against state-of-the-art CTR models, encompassing models for non-sequential, short-term, long-term, and lifelong behavior sequence modeling. For non-sequential modeling, we utilize **DNN** [8], **Wide&Deep** [7], and **PNN** [29]; for short-term behavior sequences, **DIN** [48], **DIEN** [47], and **DSIN** [13] are employed; for long-term behavior sequences, **MIMN** [26], **SIM Hard** [27], **SIM Soft** [27], **ETA** [5], **SDIM** [3], **TWIN** [4], and **MIRRN** [42] are used; and for lifelong behavior modeling, we employ **TWINv2** and adapt two quantization techniques: **RQ-VAE**[19] and **RQ-KMeans**[23]. While **RQ-VAE** and **RQ-KMeans** are originally designed for single image and text compression, respectively, we apply them to quantify user behavior sequences, with the resulting codebook vectors serving as compressed behavior proxies. All three methods utilize pre-computed representations from DIN for their clustering or quantization.

*5.1.3 Evaluation Metrics.* This research employs the Area Under the Curve (AUC) as the evaluation metric for CTR models, referencing studies like [7, 16, 29, 33, 40]. The AUC is used to assess the ranking ability of CTR models, providing a quantitative measure of their ability to rank positive instances above negative ones. Clustering performance is evaluated using the silhouette coefficient [1, 2, 11, 32], a widely adopted metric ranging from $[-1, 1]$ that quantifies intra-cluster compactness and inter-cluster separation. Higher values indicate better clustering quality.

*5.1.4 Implementation Details.* Following the dataset processing approach of [26, 42], we divide users into a training set (80%), a validation set (10%), and a test set (10%) and sort user interactions chronologically to construct behavior sequences. For user $u$ with $T$ behaviors, the first $T-1$ behaviors are used as features to predict whether the user will engage with the $T$-th item. Implementation details align with MIRRN: short-term modeling uses the most recent 100 behaviors, and long-term modeling uses the most recent 300 behaviors, with GSU retrieving a consistent number of behaviors. The final prediction layer across models is an MLP of 200×80x2, with feature dimensions set to 16. All models are trained using the Adam optimizer with batch size 256 and learning rate in {0.01, 0.001}. The number of hash bits is in {2, 3, 4, 5, 10, 16, 32, 48}. The attention dimension is in {4, 8, 16, 32, 48}. For lifelong behavior modeling, we extend the sequence length to the recent 900 behaviors. C-Former's configuration includes: cluster numbers aligned with GSU retrieval quantities, encoder layers in {1, 2, 3}, decoder layers in {1,2}, $\alpha, \beta$

[4]https://tianchi.aliyun.com/dataset/dataDetail?dataId=22482

**Table 3: Performance comparison with baselines. The best results are in bold, second-best are in underlined. * denotes statistical significance ($p < 0.05$) compared to the second-best. Due to the absence of timestamp and category in the XLong dataset, DSIN and SIM Hard cannot be implemented.**

| Model | Taobao | Alipay | Tmall | XLong |
|---|---|---|---|---|
| DNN | 0.8713 | 0.8135 | 0.8990 | 0.7130 |
| Wide&Deep | 0.8812 | 0.8223 | 0.9058 | 0.7169 |
| PNN | 0.8989 | 0.8347 | 0.9236 | 0.7214 |
| DIN | 0.9038 | 0.8459 | 0.9241 | 0.7804 |
| DIEN | 0.9061 | 0.8499 | 0.9357 | 0.7853 |
| DSIN | 0.9073 | 0.8473 | 0.9372 | – |
| SIM Hard | 0.9084 | 0.8505 | 0.9391 | – |
| SIM Soft | 0.9238 | 0.8674 | 0.9455 | 0.7908 |
| ETA | 0.9186 | 0.8608 | 0.9403 | 0.7893 |
| SDIM | 0.9084 | 0.8660 | 0.9452 | 0.7899 |
| MIRRN | 0.9319 | 0.8791 | 0.9507 | 0.7747 |
| TWIN | 0.9257 | 0.8656 | 0.9459 | 0.7915 |
| TWINv2 | 0.9309 | 0.8941 | 0.9517 | 0.7939 |
| RQ-VAE | 0.9332 | 0.8849 | 0.9512 | 0.7925 |
| RQ-KMeans | 0.9345 | 0.8815 | 0.9484 | 0.7904 |
| C-Former | **0.9457*** | **0.9055*** | **0.9604*** | **0.7982*** |

in {0.01, 0.1, 1, 10}, and $k$ is 1. Experiments are repeated 5 times to ensure stable evaluation.

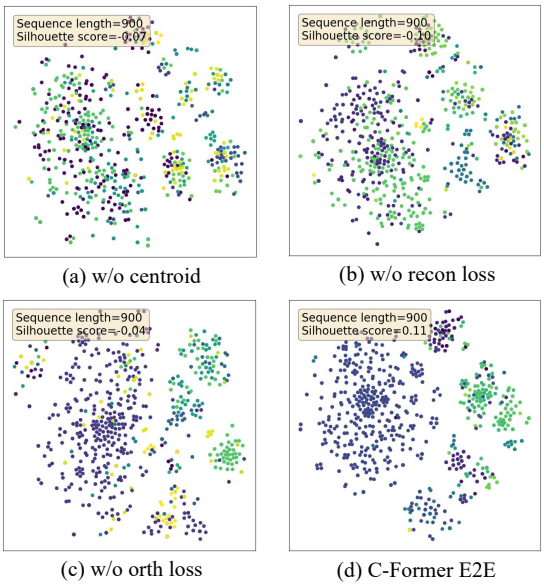
## 5.2 Overall Performance (RQ1)

We perform a comparative performance evaluation of C-Former against baselines, with the results presented in Table 3. The key observations are:

- The proposed C-Former significantly outperforms the baseline models. On four datasets, the C-Former achieves AUC improvements of 1.20%, 1.28%, 0.91%, and 0.54% over the best baseline. These results underscore the effectiveness of C-Former in modeling lifelong user behavior sequences.
- The performance of user behavior sequence modeling increases with sequence length, with lifelong modeling outperforming long-term, long-term outperforming short-term, and short-term outperforming non-sequential modeling. This demonstrates that longer sequences better capture user preferences, highlighting the significance of modeling longer user behavior sequences.
- Non-end-to-end representation learning and clustering lead to suboptimal performance. Unlike our C-Former, which jointly optimizes these two objectives, TWINv2, RQ-VAE, and RQ-KMeans rely on pre-computed recommendation representations for clustering. In these decoupled approaches, the Euclidean distance metric loses meaning as the pre-computed representations are optimized for recommendation rather than clustering objectives. Consequently, data points that are spatially proximate may lack semantic similarity, compromising recommendation performance.
- Finer-grained search mechanisms are highly effective. From SIM hard, which uses category-based hard retrieval, to SIM soft, which employs dot product relevance, to TWIN, which utilizes target attention, and finally, to C-Former, which leverages multi-layer transformers to extract centroids, performance progressively improves. This highlights the superiority of fine multi-layer transformers in C-Former.

**Table 4: Ablation study on C-Former.**

| Variants | Taobao | Alipay | Tmall |
|---|---|---|---|
| w/o denoise | 0.9384 | 0.8881 | 0.9521 |
| w/o recon loss | 0.9406 | 0.8945 | 0.9547 |
| w/o orth loss | 0.9414 | 0.8954 | 0.9453 |
| C-Former | **0.9457** | **0.9055** | **0.9604** |




(a) w/o centroid (b) w/o recon loss

(c) w/o orth loss (d) C-Former E2E

**Figure 4: T-SNE visualization of the clustering results for a specific user. The user, randomly selected from the Alipay dataset, has a behavior sequence length of 900.**

## 5.3 Ablation Study (RQ2)

To investigate the impact of different components on C-Former's performance, we design three variants stemming from C-Former: 1) **w/o denoise:** replacing denoised centroids with the refined clustering anchor points for the CTR task, 2) **w/o recon loss:** removing the reconstruction loss, equivalent to removing the decoder, and 3) **w/o orth loss:** removing the orthogonal loss. We also randomly selected a user with a behavior sequence length of 900 from the Alipay dataset and visualized the clustering results across variants using t-SNE [36]. Table 4 and Figure 4 demonstrate that:

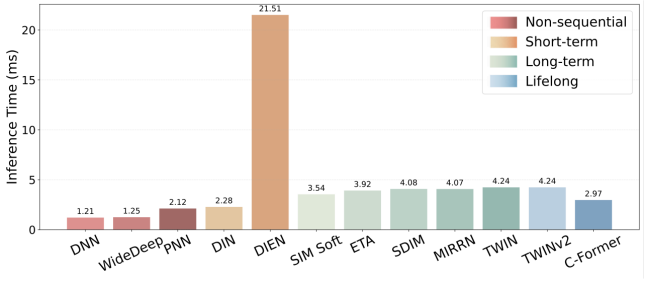



**Figure 5: The inference time on the XLong dataset.**

- Refined clustering anchors underperform centroids by failing to mitigate noise effectively. These anchors, still affected by irrelevant behaviors, harm clustering and recommendations. Unlike centroids, refined clustering anchors do not exclude extra-cluster noise, leading to poorer cluster representation for CTR tasks, which in turn weakens clustering quality through CTR supervision.
- The ablation of reconstruction and orthogonal losses deteriorates both clustering and recommendation performance. Absent these losses, centroids exhibit reduced informativeness or excessive inter-cluster proximity, degrading clustering quality. Such suboptimal centroid representations propagate to the CTR task, impairing its accuracy.
- C-Former achieves state-of-the-art performance in both clustering and recommendation tasks. Its end-to-end framework enables mutually reinforcing optimization of clustering and recommendation objectives, resulting in superior intra-cluster cohesion, distinct inter-cluster separation, and enhanced recommendation accuracy.

## 5.4 Inference Time Analysis (RQ3)

To assess C-Former's computational efficiency, we measure inference times on the XLong dataset across four model categories: (1) non-sequential (DNN, WideDeep, PNN) using average pooling on full sequences; (2) short-term (DIN, DIEN) processing 100 recent behaviors; (3) long-term (SIM Soft, ETA, SDIM, MIRRN, TWIN) handling 300 recent behaviors; and (4) lifelong (TWINv2, C-Former) analyzing 900 recent behaviors. All models use effectiveness-optimal configurations via hyperparameter tuning.

As shown in Figure 5, inference costs typically scale proportionally with sequence length: long-term models cost more than the short-term model DIN, which itself exceeds non-sequential ones. However, DIEN incurs atypically high costs due to its sequential GRU architecture limiting parallelization. Conversely, lifelong models match long-term inference times despite extended scope by using clustering to mitigate sequence-length modeling costs. Notably, C-Former replaces lifelong sequences with compact centroids generated via clustering, further reducing time complexity by 16.1% relative to the most efficient long-term baseline.

## 5.5 Performance Across Sequence Lengths (RQ4)

To assess the effectiveness of C-Former across varying sequence lengths, we conduct an evaluation using the Tmall dataset with sequence lengths of 100, 300, and 900. The findings, illustrated in Figure 6, indicate the following:

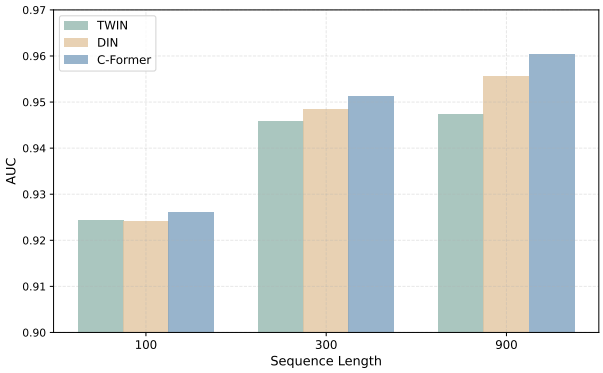



**Figure 6: The performance of models at different sequence lengths on the Tmall dataset.**

- As sequence lengths increase, the performance disparity between the GSU-based TWIN and other models becomes more pronounced. This is due to TWIN's sub-sequence retrieval mechanism, which excludes behaviors outside its retrieval scope. These excluded behaviors may contain essential user profiles or interactions related to the target, leading to greater information loss in longer sequences.
- C-Former demonstrates its most substantial improvements on lifelong user sequences. Compared to TWIN, C-Former's centroids retain comprehensive behavior information. In contrast to DIN, its denoising mechanism effectively filters out noisy behaviors. These advantages are less significant at shorter sequence lengths, where the impact of truncated contexts and noise is reduced, resulting in only slight improvements over leading baselines.

## 5.6 Sensitive Study (RQ5)

*5.6.1 number of clusters.* To assess the impact of cluster numbers on the CTR task, we conduct a series of experiments using the Taobao, Alipay, and Tmall datasets. We evaluate the AUC performance across these datasets by varying the number of clusters, testing configurations with 3, 6, 12, 24, and 48 clusters. The results, presented in Figure 7, indicate that: 1) The Taobao and Tmall datasets exhibit insensitivity to the number of clusters, whereas the Alipay dataset demonstrates sensitivity. Specifically, Alipay's performance varies significantly with an increase in cluster numbers, which may be attributed to its shorter average sequence length. 2) A cluster number of 24 provides relatively good performance across all datasets. Further increasing the number of clusters results in performance decline, possibly due to the increased complexity of the model, which can introduce more noise or lead to overfitting.

*5.6.2 number of encoder and decoder layers.* We also investigate the impact of varying the number of encoder and decoder layers on performance, with results shown in Table 5. Key conclusions are: 1) Model performance initially improves with an increasing number of encoder layers; however, beyond a critical threshold,

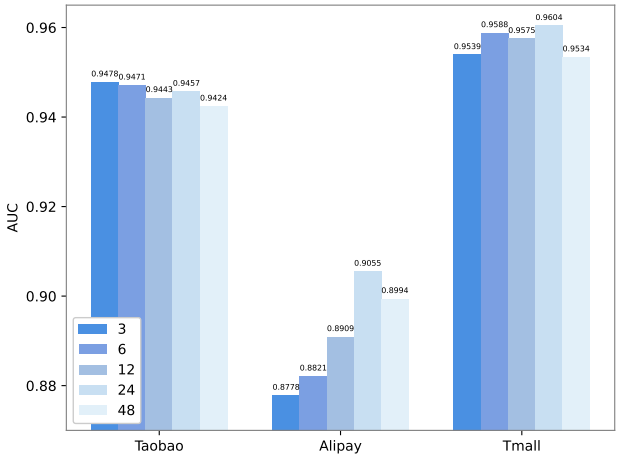



**Figure 7: The impact of the number of clusters. We assess AUC performance on the Taobao, Alipay, and Tmall datasets using cluster configurations of 3, 6, 12, 24, and 48.**

**Table 5: The impact of the number of encoder and decoder layers, where the "1*" denotes a single multi-head cross-attention layer.**

| # Encoder layers | # Decoder layers | Taobao | Alipay | Tmall |
|---|---|---|---|---|
| 1 | 1* | 0.9435 | 0.8856 | **0.9604** |
| 2 | 1* | **0.9457** | **0.9055** | 0.9588 |
| 3 | 1* | 0.9421 | 0.8673 | 0.9582 |
| 2 | 1 | 0.9420 | 0.8679 | 0.9580 |
| 2 | 2 | 0.9415 | 0.8309 | 0.9558 |

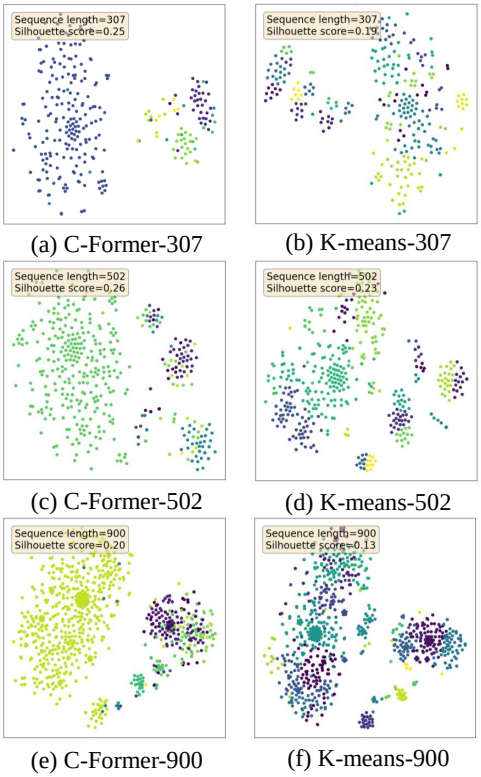



**Figure 8: T-SNE visualization of clustering users with varying behavior sequence lengths—specifically 307, 502, and 900—is evaluated using both the C-Former model and the K-means algorithm on the Alipay dataset.**

additional layers induce performance degradation, attributable to overfitting. 2) Performance consistently decreases with more decoder layers, achieving optimal results with 1* decoder layer. This aligns with the asymmetric structure principle in masked autoencoders (MAEs) [15, 21, 41, 46], where a lightweight decoder compels the encoder to develop more robust and meaningful representations. A simpler decoder enhances the encoder's learning, whereas a complex decoder may obscure the encoder's contributions. This underscores the greater importance of the encoder over the decoder in C-Former, as the encoder is responsible for extracting underlying interests from user behaviors.

## 5.7 Visualization and Case Study (RQ6)

To evaluate the clustering performance of the C-Former model, we select three users from the Alipay dataset, each exhibiting different behavior sequence lengths: 307, 502, and 900. The clustering results of their behavior embeddings, generated by both C-Former and K-means (using an equivalent number of cluster centers), are visualized in Figure 8. C-Former consistently demonstrates superior clustering effectiveness, characterized by tighter within-cluster cohesion and greater between-cluster separation, across users with varying behavior sequence lengths. It outperforms K-means in all instances, underscoring the advantage of integrating end-to-end representation learning with deep clustering, as opposed to relying solely on K-means for precomputed representations.

## 6 Conclusion

This paper proposes C-Former, a transformer-based clustering model for lifelong user behavior sequence analysis, which integrates end-to-end clustering and recommendation, and generates high-quality, cacheable centroids to minimize inference latency. The framework employs a C-Former encoder, built on multi-layer Transformers, which captures user interests by utilizing a group of learnable clustering anchor points as queries and user behavior sequence embeddings as keys and values. This design allows for the automatic aggregation of relevant behaviors and the extraction of underlying interests. A C-Former decoder reconstructs user behaviors from these anchors, followed by a denoising mechanism that filters noise via explicit cluster assignment and intra-cluster aggregation. To enhance centroid quality, a reconstruction loss preserves behavioral context, while orthogonal regularization ensures cluster diversity. By directly supervising centroid learning with recommendation signals in an end-to-end paradigm, clustering optimization aligns with downstream task objectives. Experiments on four benchmark datasets validate C-Former's effectiveness in clustering and recommendation.

## GenAI Usage Disclosure

GenAI tools are only used for polishing in writing.

## References


[1] S Aranganayagi and Kuttiyannan Thangavel. 2007. Clustering categorical data using silhouette coefficient as a relocating measure. In *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*.

[2] Adil M Bagirov, Ramiz M Aliguliyev, and Nargiz Sultanova. 2023. Finding compact and well-separated clusters: Clustering using silhouette coefficients. *Pattern Recognition* 135 (2023), 109144.

[3] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. 2022. Sampling is all you need on modeling long-term user behaviors for CTR prediction. In *CIKM*. 2974–2983.

[4] Jianxin Chang, Chenbin Zhang, Zhiyi Fu, Xiaoxue Zang, Lin Guan, Jing Lu, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, et al. 2023. TWIN: TWo-stage interest network for lifelong user behavior modeling in CTR prediction at kuaishou. In *KDD*. 3785–3794.

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

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

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

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[9] Jiaxin Deng, Shiyao Wang, Yuchen Wang, Jiansong Qi, Liqin Zhao, Guorui Zhou, and Gaofeng Meng. 2024. MMBee: Live Streaming Gift-Sending Recommendations via Multi-Modal Fusion and Behaviour Expansion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[10] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Duy-Tai Dinh, Tsutomu Fujinami, and Van-Nam Huynh. 2019. Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. In *Knowledge and Systems Sciences: 20th International Symposium, KSS 2019, Da Nang, Vietnam, November 29–December 1, 2019, Proceedings 20*. 1–17.

[12] Chao Feng, Wuchao Li, Defu Lian, Zheng Liu, and Enhong Chen. 2022. Recommender forest for efficient retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 38912–38924.

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

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16000–16009.

[16] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM conference on recommender systems*. 169–177.

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

[18] Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. 2022. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics* 11, 1 (2022), 141.

[19] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *CVPR*.

[20] Wuchao Li, Chao Feng, Defu Lian, Yuxin Xie, Haifeng Liu, Yong Ge, and Enhong Chen. 2023. Learning balanced tree indexes for large-scale vector retrieval. In *KDD*.

[21] Jihao Liu, Xin Huang, Jinliang Zheng, Yu Liu, and Hongsheng Li. 2023. MixMAE: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. In *CVPR*. 6252–6261.

[22] Qi Liu, Xuyang Hou, Haoran Jin, Zhe Wang, Defu Lian, Tan Qu, Jia Cheng, Jun Lei, et al. 2023. Deep Group Interest Modeling of Full Lifelong User Behaviors for CTR Prediction. *arXiv preprint arXiv:2311.10764* (2023).

[23] Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2024. QARM: Quantitative Alignment Multi-Modal Recommendation at Kuaishou. *arXiv preprint arXiv:2411.11739* (2024).

[24] Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. 2023. FinalMLP: an enhanced two-stream MLP model for CTR prediction. In *AAAI*.

[25] Xuan-Bac Nguyen, Duc Toan Bui, Chi Nhan Duong, Tien D. Bui, and Khoa Luu. 2021. Clusformer: A Transformer Based Clustering Approach to Unsupervised Large-Scale Face and Visual Landmark Recognition. In *CVPR*.

[26] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *KDD*. 2671–2679.

[27] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

[28] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In *SIGIR*.

[29] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.

[30] Alec Radford. 2018. Improving language understanding by generative pre-training. (2018).

[31] Steffen Rendle. 2010. Factorization machines. In *ICDM*.

[32] Ketan Rajshekhar Shahapure and Charles Nicholas. 2020. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. 747–748.

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

[34] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *CIKM*.

[35] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[37] A Vaswani. 2017. Attention is all you need. *NeurIPS* (2017).

[38] Ningning Wang, Guobing Gan, Peng Zhang, Shuai Zhang, Victor Junqiu Wei, Qun Liu, and Xin Jiang. 2022. ClusterFormer: Neural Clustering Attention for Efficient and Effective Transformer. In *ACL*. 2390–2402.

[39] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *ADKDD*.

[40] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *WWW*.

[41] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. *arXiv preprint arXiv:2205.12035* (2022).

[42] Xiang Xu, Hao Wang, Wei Guo, Luankang Zhang, Wanshan Yang, Runlong Yu, Yong Liu, Defu Lian, and Enhong Chen. 2024. Multi-granularity Interest Retrieval and Refinement Network for Long-Term User Behavior Modeling in CTR Prediction. *arXiv preprint arXiv:2411.15005* (2024).

[43] Yichen Xu, Yanqiao Zhu, Feng Yu, Qiang Liu, and Shu Wu. 2021. Disentangled self-attentive neural networks for click-through rate prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*.

[44] Yi Yang, Baile Xu, Shaofeng Shen, Furao Shen, and Jian Zhao. 2020. Operation-aware neural networks for user response prediction. *Neural Networks* 121 (2020).

[45] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*.

[46] Qi Zhang, Yifei Wang, and Yisen Wang. 2022. How mask matters: Towards theoretical understandings of masked autoencoders. *NeurIPS* (2022).

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

[48] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*.