MISS: Multi-Modal Tree Indexing and Searching with Lifelong Sequential Behavior for Retrieval Recommendation

Chengcheng Guo*
Kuaishou Inc., Beijing, China
guochengcheng03@kuaishou.com

Junda She*
Kuaishou Inc., Beijing, China
shejunda@kuaishou.com

Kuo Cai
Kuaishou Inc., Beijing, China
caikuo@kuaishou.com

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

Qigen Hu
Kuaishou Inc., Beijing, China
huqigen03@kuaishou.com

Qiang Luo†
Kuaishou Inc., Beijing, China
luoqiang@kuaishou.com

Kun Gai
Unaffiliated, Beijing, China
gai.kun@qq.com

Guorui Zhou†
Kuaishou Inc., Beijing, China
zhouguorui@kuaishou.com

Abstract

Large-scale industrial recommendation systems typically employ a two-stage paradigm of retrieval and ranking to handle huge amounts of information. Recent research focuses on improving the performance of retrieval model. A promising way is to introduce extensive information about users and items. On one hand, lifelong sequential behavior is valuable. Existing lifelong behavior modeling methods in ranking stage focus on the interaction of lifelong behavior and candidate items from retrieval stage. In retrieval stage, it is difficult to utilize lifelong behavior because of a large corpus of candidate items. On the other hand, existing retrieval methods mostly relay on interaction information, potentially disregarding valuable multi-modal information. To solve these problems, we represent the pioneering exploration of leveraging multi-modal information and lifelong sequence model within the advanced tree-based retrieval model. We propose **M**ulti-modal **I**ndexing and **S**earching with lifelong **S**equence (**MISS**), which contains a multi-modal index tree and a multi-modal lifelong sequence modeling module. Specifically, for better index structure, we propose multi-modal index tree, which is built using the multi-modal embedding to precisely represent item similarity. To precisely capture diverse user interests in user lifelong sequence, we propose collaborative general search unit (Co-GSU) and multi-modal general search unit (MM-GSU) for multi-perspective interests searching. Online experiments have demonstrated the effectiveness of the proposed method.

CCS Concepts

• **Information Systems → Recommendation Systems.**.

*Equal contribution.
†Corresponding author.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Keywords

Recommender Systems, Multi-Modal, Long Sequential User Behavior, Tree-Based Learning

ACM Reference Format:
Chengcheng Guo, Junda She, Kuo Cai, Shiyao Wang, Qigen Hu, Qiang Luo†, Kun Gai, and Guorui Zhou. 2018. MISS: Multi-Modal Tree Indexing and Searching with Lifelong Sequential Behavior for Retrieval Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

To balance efficiency and effectiveness, a cascade framework, which consists of retrieval (also called matching, recall, *etc.*) and ranking stages, has been widely adopted in modern recommendation systems [7]. Within this framework, retrieval stage is crucial in distinguishing candidates from the vast corpus but given the least time. DSSM[11] is a classic dual-tower architecture, which employs the Approximate Nearest Neighbor (ANN) method for efficient item retrieval. Within the dual-tower paradigm, user representation and item representation can only interact at a later stage. For early interaction, a line of work seeks to design index structure for candidate retrieval. Tree-based Deep Models (TDM[28], JDM[27], BSAT[29]) propose a tree-based index to hierarchically retrieve candidates. NANN[5] resorts to HNSW[20], a graph-based index. Deep Retrieval (DR)[8] defines items into "paths". Recently, Streaming VQ[1] utilizes vector quantization as a new index structure. With the index structure, complicated models originally designed for the ranking stage can be used in the retrieval stage for representations interaction between users and items.

Among complicated models in the ranking stage, lifelong sequential behavior modeling methods (e.g. SIM) are effective. The interactions of lifelong sequence and candidate items from retrieval stage using complicated models are effective to model user interests. SIM[22] is a search-based method, which introduces a General Search Unit (GSU) for behavior searching and an Exact Search Unit (ESU) for precise relationship modeling. TWIN[3] extend the target

attention structure to GSU, thus addressing the inconsistency between GSU and ESU. In retrieval stage, a large corpus of candidate items makes it difficult to utilize the lifelong sequence.

The above methods depend on behavior information, potentially disregarding valuable multi-modal information. With the significant evolution of multi-modal large models, the potential of multi-modal information in the recommendation system has received great attention[15, 16]. AlignRec[17] trains multi-modal embeddings using visual-text alignment task and collaborative filtering task. Sheng et al. [24] proposes semantic-aware contrastive learning to pre-train multi-modal embeddings. QARM[18] aligns multi-modal embeddings with item-item relationships in pre-training and leverages quantitative code of multi-modal embeddings for recommendation. Above all, existing multi-modal work mostly concentrates on the learning of multi-modal embeddings.

To tackle these problems, we propose **M**ulti-modal **I**ndexing and **S**earching with lifelong **S**equence (**MISS**), a pioneer model that introduces lifelong sequence model and multi-modal information into tree-based model. To be specific, MISS contains two components: a multi-modal index tree and a multi-modal lifelong sequential behavior modeling module. Since this paper does not focus on the training of multi-modal embeddings, we directly use item alignment[18] to train multi-modal embeddings, which contain content and interaction information. As for the index tree, it is essential to organize similar items in the same subtree[28]. Thus, we construct the tree using multi-modal embeddings to precisely reflect item similarity. As for modeling lifelong behavior, it is crucial to extract diverse interests in user lifelong sequence. To achieve this goal, we propose a collaborative general search unit (Co-GSU) and a multi-modal general search unit (MM-GSU), which search user interests with collaborative and multi-modal information. In fact, MISS has been serving in Kuaishou's recommendation system, leading to remarkable user engagement gain. Our contributions can be summarized as follows:

- To the best of our knowledge, our work represents the pioneering exploration of leveraging multi-modal information within advanced retrieval models.
- We propose a multi-modal tree-based deep model, which contains a multi-modal index tree and a lifelong sequence learning modulde with Co-GSU and MM-GSU.
- We conduct extensive experiments to demonstrate the effectiveness of our method. Our method is also validated through online A/B test, showing promising results.

2 Related Works

2.1 Indexing

In the retrieval stage, indexing schemes are essential for effectively organizing and retrieving large-scale items[10]. DSSM[11] is so-called 'two tower model', which searches by Approximate Nearest Neighbor (ANN) method. Tree-based Deep Models (TDM[28], JDM[27], BSAT[29]) propose tree structures to hierarchically search candidates from coarse to fine and make decisions for each user-node pair. NANN[5] resorts to HNSW[20] for candidate searching. To avoid the Euclidean space assumption in the ANN algorithms, Deep Retrieval (DR)[8] defines items into 'paths' and uses beam search to shrink candidates layer by layer. Recently, Streaming VQ[1] utilizes vector quantization as a novel index structure.

2.2 Lifelong Sequential Behavior Modeling

User Long Sequence Modeling is vital for capturing user interests [9, 22]. MIMN[21] integrates user behaviors learning with serving systems. SIM [22] introduces a General Search Unit (GSU) for behavior retrieval and an Exact Search Unit (ESU) for precise relationship modeling. EAT [4] uses locality-sensitive hashing for item embeddings and Hamming distance for retrieval. SDIM [2] samples items with the same hash signature, aggregating these via ESU to derive user interests. TWIN [3] extends the target attention structure to GSU and synchronizes embeddings and attention parameters between ESU and GSU.

2.3 Multi-Modal Recommendation

DVBPR[12] jointly trains a CNN visual encoder with the Matrix Factorization task. BM3[26] leverages self-supervised learning to align both the inter-modality and intra-modality representations within the collaborative filtering task. AlignRec[17] pre-trains the visual-text alignment task and then aligns the multi-modal representations and the ID representations supervised by collaborative filtering task. Sheng et al. [24] proposes semantic-aware contrastive learning in the pre-training stage and extracts features using SimTier and MAKE based on fixed multi-modal representations. QARM[18] aligns multi-modal representation with downstream business-specific item-item relationships in pre-training and leverages quantitative code mechanisms for end-to-end training in recommendation models.

As mentioned above, existing work mostly focuses on the pre-training stage to learn better multi-modal representations. However, how to effectively utilize multi-modal information in the retrieval recommendation has rarely been explored.

3 Preliminary

Problem Definition. Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and videos, respectively, with the size of the video set given by $|\mathcal{V}|$. During the retrieval phase, for any given user $u \in \mathcal{U}$, we aim to extract a small subset of videos $\mathcal{V}_u \subset \mathcal{V}$ that align with user interests, satisfying $|\mathcal{V}_u| \ll V$. In our model $\mathcal{M}$, we employ a multi-objective learning approach to optimize it. For videos v watched by user u, user feedback—likes, comments, playing completions—is captured in a binary vector $\{y_1, \ldots, y_T\} \in \mathbb{R}^T$, where each $y_i \in \{0, 1\}^V$. The model learns from these feedbacks across T different objectives. Each video is also represented by a multi-modal feature vector m. Our model is structured as a tree, comprising an index tree $\mathcal{T}$ and a node estimator $\mathcal{F}_\theta$. Following [6], we denote the training dataset for task t as $\mathcal{D}_{tr}^t$ and the testing dataset as $\mathcal{D}_{ts}$.

Multi-Modal Embedding. To introduce behavior information into content-based representations, we follow the item alignment mechanism[18] to supervise the fine-tuning of content-based embedding with knowledge from retrieval models. Specifically, we select item pairs (i, j) with high similarity from our item2item retrieval model as the data source $\mathcal{D}$, then train a multi-modal representation model (MRM) with pure content-based inputs (images and text) and interaction-based supervision signals. For a random

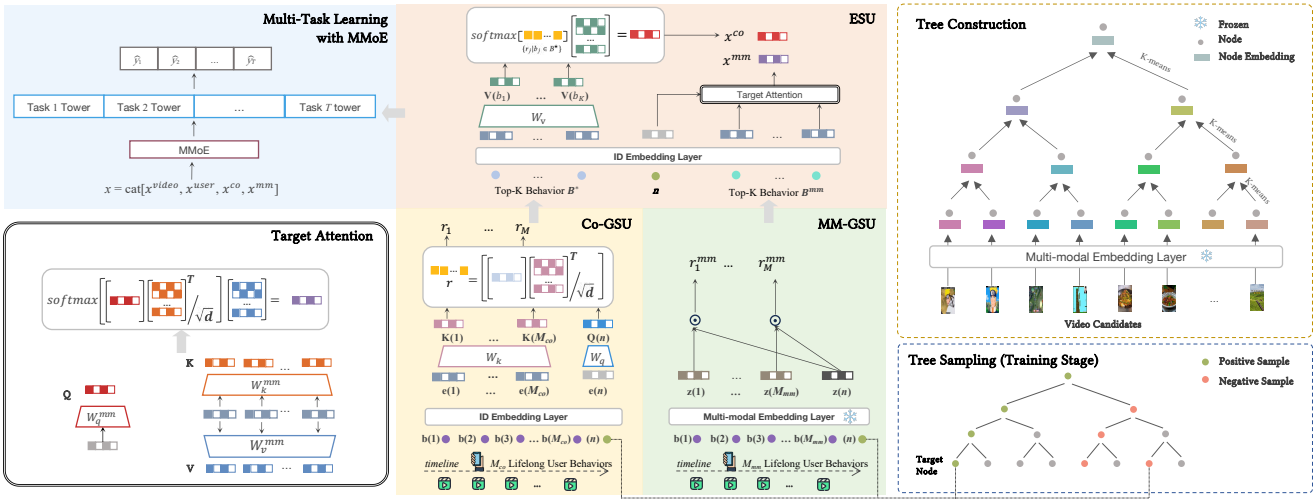

Figure 1: The overall framework of MISS. a) Tree Construction: the multi-modal index tree is first built based on multi-modal embeddings using k-means. b) Tree Learning: given a positive or negative sample and lifelong user behaviors, the Co-GSU and MM-GSU would calculate the relevant scores r_i and r_i^{mm}, respectively. Top-K behaviors are selected as the inputs of ESU. In ESU, target attention is performed to get two representations x^{co} and x^{mm}. Finally, multiply representations are sent into MMoE module for 'pxtr' prediction.

data batch in $\mathcal{D}$, we select in-batch negative samples $\mathcal{J}$ for training.

$$c_i = \textbf{MRM}(c_i^{text}, c_i^{image}),\ c_j = \textbf{MRM}(c_j^{text}, c_j^{image}), \tag{1}$$

$$\mathcal{L}_{IA} = \textbf{InfoNCE}(c_i, c_j, \mathcal{J}), \tag{2}$$

where c_i and c_j are the generated multi-modal embeddings, and **InfoNCE** pulls positive samples closer while pushing negatives away. As a result, the so-called multi-modal embedding contains both content modality and interaction modality.

Index Tree. Let $\mathcal{T}$ be a binary tree of height H, with the root at level 0 and leaves at level H. We denote nodes at level h as $\mathcal{N}_h$ and all nodes as $\mathcal{N} = \bigcup_{h=0}^{H} \mathcal{N}_h$. Each leaf node corresponds uniquely to a video, establishing a bijective mapping $\pi : \mathcal{N}_H \to \mathcal{V}$. For each node $n \in \mathcal{N}$, we denote leaf nodes in its subtree by $\textbf{Leaf}(n)$ and its ancestor at level h as $\textbf{Anc}(n; h)$.

4 Methods

4.1 Multi-modal Index Tree

In this section, we build the multi-modal index tree $\mathcal{T}$ using our multi-modal embedding because of its outstanding properties. As mentioned in TDM[28], the tree should represent hierarchical information of user interests, and it is natural to build the tree in a way that similar items are organized in close position. Recall that the multi-modal embeddings are trained to pull similar items closer using content representation. Thus, the multi-modal embeddings are equipped to represent multidimensional similarity, both in terms of content and interactions. Experimental results show the effectiveness of our multi-modal index tree.

Tree Construction. With the multi-modal embedding, we construct a tree using the k-means algorithm. At each step, items are divided into two clusters according to their multi-modal embeddings. Subsequently, similar operations are conducted within each of the clusters. The recursion stops when only one item is left and a binary tree could be constructed in such a top-down way.

Each non-leaf node is a cluster center and its embedding is the mean pooling of the multi-modal embedding of the leaf node in its subtree. Formally, the multi-modal embedding for each node is as follow:

$$z_i = \begin{cases} c_{\pi(i)}, & \text{if } i \in \mathcal{N}_H, \\ \textbf{MeanPooling}(\{c_{\pi(j)} | j \in \textbf{Leaf}(i)\}), & \text{else.} \end{cases} \tag{3}$$

where $\pi(i)$ is the corresponding video of node i, $\mathcal{N}_H$ is the set of leaf node, **MeanPooling** is the mean pooling operation, $\textbf{Leaf}(i)$ is the leaf node set of the subtree of node i.

Learning of Tree Model. To make the tree a max-heap[28], we first give each node an ID embedding and train the model $\mathcal{F}_\theta : \mathcal{V} \times \mathcal{N} \to \mathbb{R}$ with node-wise task. For any instance $(u, \mathbf{y})$, a pseudo label $\tilde{y}_n \in \{0, 1\}$ is defined for each node $n \in \mathcal{N}$ to represent the existence of relevant targets on the subtree of n, i.e.,

$$\tilde{y}_n = \mathbb{I}(\{\sum_{i \in \textbf{Leaf}(n)} y_{\pi(i)} \geq 1\}), \tag{4}$$

With the pseudo labels, we can train the node estimator $\mathcal{F}_\theta$ level by level and negative sampling would be used for each level. For level h of the tree, the positive target set is defined as $\mathcal{S}_h^+(y) = \{n : \tilde{y}_n = 1, n \in \mathcal{N}_h\}$. $\mathcal{S}_h^-(y)$ contains several negative samples from $\{n : \tilde{y}_n = 0, n \in \mathcal{N}_h\}$. The sample set for level h is defined as $\mathcal{S}_h(y) = \mathcal{S}_h^+(y) \bigcup \mathcal{S}_h^-(y)$. Formally, the training loss is as follow:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{(u,y) \in \mathcal{D}_{tr}^t} \sum_{h=1}^{H} \sum_{n \in \mathcal{S}_h(y)} \mathcal{L}_{BCE}(\tilde{y}_n, \hat{y}_n^t), \tag{5}$$

$$\mathcal{L}_{BCE}(\tilde{y}_n, \hat{y}_n^t) = -\tilde{y}_n log y - (1 - \tilde{y}_n) log(1 - \hat{y}_n^t), \tag{6}$$

$$\hat{y}_n^t = \mathcal{F}_\theta(u, n; t), \tag{7}$$

where $\mathcal{F}_\theta$ is the node estimator. $\mathcal{F}_\theta$ contains a multi-modal lifelong sequence learning module and a multi-task learning module, which are introduced in Section 4.2 and Section 4.3 respectively.

Unlike TDM[28], we would not build a new index tree using the trained ID embeddings, because the ID embeddings are trained just with interaction data and the absence of content information would lead to suboptimal results. The experimental results have also demonstrated the effectiveness of our multi-modal index tree.

Inference of Tree Model. When inferring, we use the beam search to sample multiply nodes. For any testing instance $(u, \mathbf{y}) \in \mathcal{D}_{ts}$, suppose $\mathcal{B}_h^{(K)}(u)$ denotes the node set retrieved at level h through beam search and $K = |\mathcal{B}_h^{(K)}(u)|$ denotes the beam size, the beam search process is defined as:

$$\mathcal{B}_h^{(K)}(u) \in \underset{n \in \tilde{\mathcal{B}}_h(u)}{\text{argTopK}}\, p_{\mathcal{F}}(\tilde{y}_n = 1|u), \tag{8}$$

where $\tilde{\mathcal{B}}_h(u) = \bigcup_{n \in \mathcal{B}_{h-1}^{(K)}(u)} C(n)$, $p_{\mathcal{F}}(\tilde{y}_n = 1|u)$ is the possibility from node estimator $\mathcal{F}$. By applying Eq.8 recursively until $h = H$, beam search retrieves the set containing K leaf nodes, denoted by $\mathcal{B}_H^{(K)}(u)$. Let $m \leq K$ be the retrieval number, the retrieval target set is defined as follow:

$$\hat{\mathcal{V}}_u = \{\pi(b) : n \in \mathcal{B}_H^{(m)}(u)\}, \tag{9}$$

$$\mathcal{B}_H^{(m)}(u) = \underset{n \in \mathcal{B}_H^{(K)}(u)}{\text{argTopm}}\, p_{\mathcal{F}}(\tilde{y}_n = 1|u), \tag{10}$$

4.2 Multi-Modal Searching with Lifelong Sequence

In this section, we introduce user lifelong behavior sequence in the retrieval stage for better user modeling. Advanced researches[21, 22] show that considering long-term historical behavior sequences in user interest modeling can significantly improve prediction performance of XTR". Although a longer user behavior sequence introduces useful information about user interest, it contains massive noise at the same time[9, 22, 23]. To mitigate the influence of noise and precisely capture user interest, we introduce two kinds of general search unit (GSU): multi-modal general search unit (MM-GSU) and collaborative general search unit (Co-GSU). MM-GSU and Co-GSU would retrieve relevant behavior based on multi-modal information and collaborative information, respectively. Then an exact search unit (ESU) models the precise relationship between the candidate TDM node and the retrieved behaviors.

Co-GSU. Given the list of user behavior $B = [b_1; b_2; ...; b_{M_{co}}]$ and a candidate node n from $\mathcal{T}$ (either virtual node or real item node), each b_i and n are first denoted as one-hot vectors and then embedded into low-dimensional vectors $E = [e_1; e_2; ...; e_{M_{co}}]$ and e_n through the ID embedding layer. After that, we take advantage of target attention[25] to calculate the relevant score:

$$r_i = (W_q e_n)^T \cdot (W_k e_i)/\sqrt{d}, \tag{11}$$

where W_q and W_k are the parameters of query matrix and key matrix, e_i and e_n are denoted as the ID embeddings of the i-th behavior b_i and the candidate node n respectively, d is the dimension of ID embeddings. After that, the Top-K relevant behaviors are selected as a **Co**ollaborative **S**ub user **B**ehavior **S**equence (Co-SBS) $B^\star$.

MM-GSU. Given the list of user behavior $B = [b_1; b_2; ...; b_{M_{mm}}]$ and a candidate node n from $\mathcal{T}$ (either virtual node or real item node), the lookup operation converts each b_i and n into multi-modal embeddings $Z = [z_1; z_2; ...; z_{M_{mm}}]$ and z_n. It is worth noticing that the multi-modal embeddings are well pre-trained and frozen in MM-GSU. We then use the multi-modal embeddings to calculate the relevant score r_i for each behavior b_i. Unlike the Co-GSU, we directly utilize the multi-modal embedding of candidate node as query and multi-modal embedding of each behavior b_i as key. The reason is that the multi-modal embeddings are trained to pull similar items closer and have formed a Euclidean space which suits calculating relevant scores.

Formally, the relevant score is calculated as $r_i^{mm} = (z_n)^T \cdot z_i$, where z_i and z_n are denoted as the multi-modal embeddings of the i-th behavior b_i and the candidate node n respectively. After that, the Top-K relevant behaviors are selected as a **M**ulti-**M**odal **S**ub user **B**ehavior **S**equence (MM-SBS) $B^{mm} = [b_1, b_2, ..., b_K]$.

ESU. With the MM-SBS and Co-SBS, the exact search unit (ESU) applies target attention between the candidate node and the two subsequences, respectively. It is worth noticing that the attention module of MM-SBS, Co-SBS and Co-GSU shares parameters.

Recalling that we have calculated the relevant scores in Co-GSU, we can reuse the scores to reduce the computation. We directly compute ***softmax*** on the scores of Co-SBS to get the attention scores.

$$a_i = softmax(\{r_j | b_j \in B^\star\})_i,\ x^{co} = \sum_{b_i \in B^\star} a_i W_v e_i. \tag{12}$$

where W_v are the parameters of projection matrices.

As for MM-SBS $B^{mm} = [b_1, b_2, ..., b_K]$, each b_i and n are first denoted as one-hot vectors and then embedded into low-dimensional vectors $E^{'} = [e_1; e_2; ...; e_K]$ and e_n through the ID embedding layer.

$$a_i = softmax(\{(W_q^{mm} e_n)^T \cdot (W_k^{mm} e_j)/\sqrt{d} | e_j \in E^{'}\})_i, \tag{13}$$

$$x^{mm} = \sum_{b_i \in B^{mm}} a_i W_v^{mm} e_i. \tag{14}$$

where W_q^{mm}, W_k^{mm} and $W_v^{mm}$are the parameters of projection matrices, which are not shared parameters with Co-SBS.

4.3 Multi-Task Learning with MMoE

The collaborative behaviors representation and multi-modal behaviors representation are subsequently concatenated with user feature and video feature of target item to construct the input of Multi-gate Mixture-of-Experts module (MMoE)[19]. In MMoE module, we incorporate L experts and introduce a separate gating network g^t for each objective t. Then individual prediction $\hat{y}^t$ for each objective follows the corresponding objective-specific tower h^k. Formally, the output of objective t is

$$\hat{y}_n^t = h^t(f^t(x)),\ f^t(x) = \sum_i^L g^t(x)_i f_i(x), \tag{15}$$

$$x = concat(x^{video}, x^{user}, x^{co}, x^{mm}). \tag{16}$$

where f_i is the expert network, $g^t(x)$ is the gated network, $g^t(x)_i$ is the i-th logit of $g^t(x)$, h^t is the tower network of objective t.

With the predicted possibility $\hat{y}_n^t$, the final loss can be calculated with Eq. 5-6. During inference, we would use the mean of the prediction scores of all objectives as the final prediction score.

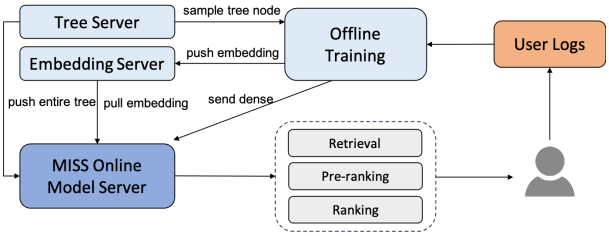

Figure 2: System deployment of the proposed method.

5 System Deployment

We deploy MISS on the retrieval stage of Kuaishou, serving 400 million daily active users.In this section, we introduce the deployment of MISS at Kuaishou. Details of our system architecture is shown in Figure 2.

Offline Training. Our proposed MISS is trained on a large-scale distributed learning system of Kuaishou. Every day, Hundreds of millions of users watch short videos at Kuaishou APP, generating tens of billions of training logs. Before training, Tree Server performs kmeans construction on the pre-trained video multimodal embedding. During the training phase, the trainer requests Tree Server to pull the tree node representation and update the sparse features to the embedding server.

Online Serving. MISS Online Model Server loads the neural network parameters and the entire tree structure sent by the Tree Server at the initialization stage, and performs BeamSearch to retrieve video results in the online stage. MISS takes effect in the retrieval stage of the Kuaishou recommendation system, and is filtered with other retrieval channels in the Pre-ranking and Ranking stages, and finally displayed to users.

6 Experiments

To verify the effectiveness of our method, we conduct experiments to answer the following research questions:

RQ1 : How effective is MISS compared to the state-of-the-art models in the industry?
RQ2 : What is the effect of each component of the proposed MISS?
RQ3 : How does the length of user lifelong sequence affect the performance of MISS ?
RQ4 : Can our MM-GSU pay more attention to long-term interests? What is the search mechanism of Co-GSU and MM-GSU?
RQ5 : Can our MISS drive the growth of online metrics?

6.1 Experimental Setting

6.1.1 Baselines. We compared with five state-of-the-art models used in industry recommendation systems.

- SASRec[13] is a sequential recommendation model which utilizes self-attention to capture the long-term preferences of users. In our system, SASRec is combined with two-tower paradigm.
- TDM[28] propose tree structures to hierarchically search candidates from coarse to fine and make decisions for each user-node pair using a deep model. In our recommendation system, the deep model of TDM is SIM[22].
- TDM+MMoE is the combination of TDM and MMoE.
- NANN[5] utilizes HNSW[20] for candidates searching.
- Kuaiformer[14] proposes a transformer-based two tower framework for retrieval. It is a strong baseline in Kuaishou's online recommendation system.

6.1.2 Metrics. To evaluate the performance of retrieval models, we use a widely adopted metric: Recall@K. For tree-based models, since they would retrieve intermediate nodes during beam search, we propose hierarchical recall to further estimate the effectiveness of the retrieval non-leaf nodes. Hierarchical recall H@hRecall@K is defined as Recall@K at level h of index tree. For user u, as stated previously, their interacted item set is denoted by $\mathcal{V}_u$ and the retrieval node set at level h during beam search is denoted by $\mathcal{B}_h^{(K)}(u)$. Suppose $\mathcal{V}_u^h = \{\mathbf{Anc}(n;h)|n \in \mathcal{V}_u\}$ to be the set of ancestor nodes at level h of $\mathcal{V}_u$, the hierarchical recall H@hRecall@K is defined as follow:

$$\text{H@hRecall@K}(u, \mathcal{V}_u) = \frac{|\mathcal{V}_u^h \bigcap \mathcal{B}_h^{(K)}(u)|}{|\mathcal{V}_u^h|}, \tag{17}$$

6.1.3 Dataset. We tested the model on mass real industrial data. Kuaishou is one of the world's largest short video apps with over **400 million** daily active users. Kuaishou generates a massive amount of data to support model training and evaluation, with more than **50 billion** user logs available per day. To handle such large-scale training data, all models (including the baseline) are optimized using an online learning paradigm.

To further validate the effectiveness of the model in real industrial scenarios, we will present the results of online A/B tests conducted with real users later in this paper.

6.1.4 Details. All models, including baselines, are trained in a streaming paradigm. For tree models, we construct a binary tree with 22 levels and apply the following beam search strategy. In the main results, Co-GSU has a length of 2000, MM-GSU has 4000 (see details in Sec. 6.4.1), and ESU has a length of 50.

6.2 Overall Performance (RQ1)

6.2.1 Final retrieval results. To demonstrate the effectiveness of MISS in retrieval recommendation, we compare it with five state-of-the-art models, which are practical in the actual industry. We set sequence length of MM-GSU as 2000 (MISS 2k) and 4000 (MISS 4k). The experiment is carried out on Kuaishou's real industrial data with recall as the metric. We randomly sample multiply testing datasets and calculate the mean and variance. The result is reported in Table. 1. From the result, we find that MISS is state-of-the-art, while "TDM+MMoE" is the second best. Compared with the second-best results, our method achieves an improvement of 37.93%, 30.77% and 47.37% in recall@800, recall@600 and recall@400, respectively. On average, our method achieves an improvement of 38.69%, which evidently confirms the effectiveness of our method.

Table 1: Final retrieval results using recall as metric. The best and second-best results highlighted in bold font and underlined.

metric	method							improvement
	SASRec	NANN	Kuaiformer	TDM	TDM+MMoE	MISS (2k seq)	MISS (4k seq)	
Recall@800	0.20±0.01305	0.23±0.01875	0.26±0.01531	0.22±0.02420	0.29±0.01558	0.38±0.01698	**0.40±0.01232**	37.93%
Recall@600	0.18±0.00585	0.20±0.01301	0.23±0.01111	0.19±0.02575	0.26±0.01451	0.33±0.01373	**0.34±0.01607**	30.77%
Recall@400	0.15±0.00516	0.15±0.01972	0.19±0.02039	0.14±0.02796	0.18±0.02562	0.26±0.02220	**0.28±0.01570**	47.37%

Table 2: Retrieval results of intermediate nodes selected during beam search. We analyze the levels 13, 16 and 19 using hierarchical recall metric. The best and second-best results in each column are highlighted in bold font and underlined.

metric	TDM	TDM+MMoE	MISS (2k seq)	MISS (4k seq)
H@13Recall@800	0.76±0.01936	0.76±0.01504	0.83±0.01703	**0.85±0.01366**
H@16Recall@800	0.38±0.02100	0.42±0.01310	0.60±0.01223	**0.64±0.01003**
H@19Recall@800	0.26±0.02132	0.32±0.01507	0.46±0.01775	**0.51±0.01426**
H@13Recall@600	0.66±0.01922	0.70±0.01015	0.80±0.01526	**0.81±0.00975**
H@16Recall@600	0.31±0.02091	0.38±0.01701	0.56±0.01008	**0.58±0.00961**
H@19Recall@600	0.21±0.01979	0.28±0.01014	0.42±0.01439	**0.46±0.01477**
H@13Recall@400	0.55±0.02721	0.58±0.02353	0.70±0.01380	**0.72±0.01091**
H@16Recall@400	0.24±0.02563	0.28±0.03447	0.46±0.01564	**0.49±0.01735**
H@19Recall@400	0.16±0.02603	0.20±0.02357	0.34±0.01865	**0.38±0.01665**

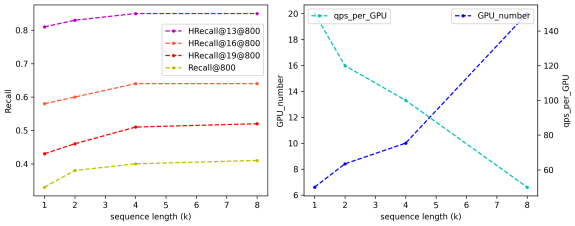

Figure 3: In-depth analysis of the length of user behavior sequence. The left figure shows the relation between recall and sequence length. The right figure shows the relation between computation resource and sequence length.

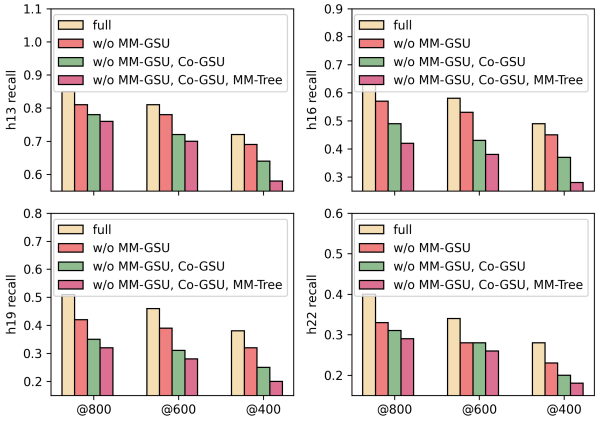

Figure 4: Ablation study of four variants using hierarchical recall metric.

6.2.2 Intermediate retrieval results. To demonstrate effectiveness, we evaluate model performance using hierarchical recall as described in Sec. 6.1.2. Beam search method in Sec. 6.1.4 allows us to analyze hierarchical recall performance, specifically the H@hRecall@K metric at layers 13, 16, and 19. As shown in Table 2, our method significantly outperforms baseline models. Notably, for H@13Recall@800, H@16Recall@800, and H@19Recall@800, our model achieves hierarchical recall increases of 11.84%, 52.38%, and 59.38%, respectively.

6.3 Ablation Study (RQ2)

In this section, we examine the impact of our three key modules: MM-GSU, Co-GSU and the multi-modal index tree. We progressively remove each module from the full model and at each step we can verify the impact of the module. To be specific, we have the following four versions: 1) full model; 2) version1: full model w/o MM-GSU; 3) version2: version1 w/o Co-GSU, 4) version3: version2 w/o multi-modal index tree. We build an index tree using ID embedding as an alternative.

We evaluate each model variant using hierarchical recall of levels 13, 16, 19 and 22 (leaf). The results are summarized in Figure. 4. From the results, it is obvious that removing each module would lead to performance degradation at the four levels, which demonstrates the effectiveness of each module.

6.4 In-depth Analysis (RQ3-RQ4)

6.4.1 Analysis of Sequence Length. We analyze the relationship between sequence length, performance, and computational resource requirements. Specifically, we assess the number of GPUs needed (*GPU_number*) and the queries per second each GPU can handle (*qps_per_GPU*). We tested four sequence lengths of MM-GSU: 1k, 2k, 3k, and 4k, with results summarized in Fig. 3.

Notably, there is a positive correlation between sequence length and both performance and GPU resource usage, indicating a trade-off between computational efficiency and model performance. For instance, increasing the sequence length from 4k to 8k yields a slight performance increase (recall@800 from 0.4 to 0.41) but significantly raises resource demand (*GPU_number* from 10 to 20, *qps_per_GPU* from 100 to 50). Thus, we ultimately set the sequence length to 4k to balance efficiency and performance.

6.4.2 Analysis of Attention Scores. We visualized the attention scores of Co-GSU (sequence length 2000) and MM-GSU (sequence length 4000) during the general search unit phase of MISS. These scores were compared with the first block self-attention scores from SASRec[13] (sequence length 4000), as shown in Fig. 5. Each search method sampled real viewing histories from 20 users, displayed across different rows of the attention map. The leftmost sections represent older viewed items, while the rightmost sections indicate more recent ones.

Notably, items with high attention scores in SASRec are mainly clustered on the right side of the figure, reflecting recent user views, while long-term interests are largely overlooked. In contrast, the

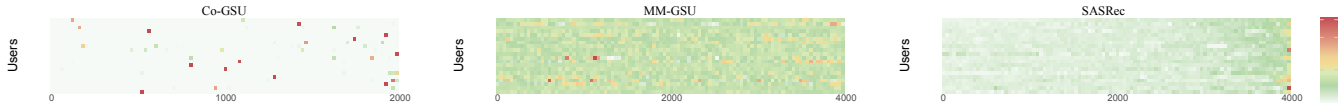

Figure 5: Softmax attention scores during the General Search Unit (GSU) phase for Co-GSU (sequence length of 2000) and MM-GSU (sequence length of 4000), juxtaposed with the self-attention score of SASRec (sequence length of 4000).

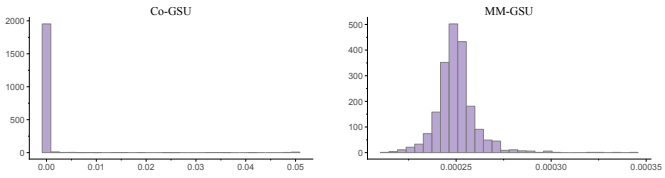

Figure 6: Histograms of Softmax attention scores. Very few items in user sequence can reach high attentions scores in Co-GSU, while in MM-GSU, various locations within the sequence have more equal opportunity to achieve high scores.

Table 3: Online A/B Test Result. Confidence intervals (CI) are calculated with 0.05 significance level.

	Total App Usage Time	Total App Usage Time (CI)	App Usage Time Per User	App Usage Time Per User (CI)	Video Watch Time
NANN	+0.103%	[0.04%, 0.17%]	+0.081%	[0.02%, 0.14%]	+0.234%
Kuaiformer	+0.118%	[0.03%, 0.21%]	+0.080%	[0.01%, 0.15%]	+0.286%
TDM (w/o MM tree)	+0.121%	[0.03% , 0.21%]	+0.085%	[0.02% , 0.15%]	+0.132%
TDM (w/o MM searching)	+0.166%	[0.02%, 0.31%]	+0.136%	[0.02%, 0.25%]	+0.387%
Poposed	**+0.248%**	**[0.16%, 0.34%]**	**+0.212%**	**[0.14%, 0.28%]**	**+0.584%**

GSU, by incorporating query items, enables search results to emerge from various positions within the sequence, rather than being restricted to the most recently viewed items. This feature allows the GSU search mechanism to better utilize the rich information in users' long sequences during the recall phase.

The results suggest that the GSU's search mechanism, particularly in utilizing long-term user sequences, outperforms the dual-tower model's self-attention. This is mainly due to the lack of personalized queries in the dual-tower self-attention, which limits its ability to effectively search for distant tokens.

6.4.3 Analysis of Two GSU. When comparing MM-GSU with Co-GSU, it is evident that Co-GSU exhibits significant score disparities within a single sequence (see Fig. 5), resulting in high attention score positions being relatively sparse. As shown in Fig. 6, the distribution of softmax attention scores for Co-GSU is highly uneven, with only a small number of tokens achieving high attention scores.

In contrast, MM-GSU displays a denser distribution of high attention score locations (see Fig. 5), suggesting that various locations within the sequence have more equal opportunity to achieve high scores. This feature enables multi-modal retrieval to provide a more equitable chance for videos within the sequence to be searched, thereby improving the overall accuracy of the model.

More generally, we analyzed the overlap rate of retrieval results between the real samples in Co-GSU and MM-GSU, which was found to be **only 13%**. This indicates that the two systems complement each other effectively.

6.5 Online A/B Test Result (RQ5)

To evaluate the online performance of MISS , we conduct strict online A/B tests on Kuaishou's video recommendation scenarios. For each recall model, we evaluated its performance within the single-page recommendation system of Kuaishou Lite. Each experiment was conducted over a period of seven days, involving 5% of the total user. We compared the experimental group against the baseline group in terms of Total App Usage Time and App Usage Time Per User, assessing the growth rates and their corresponding 95% confidence intervals. Additionally, we analyze the increase in video watch time.

Shown in Tab. 3, the experimental results indicate that the recall model incorporating proposed multimodal tree construction and multimodal search modules achieved the best performance. Specifically, we observed a growth of **0.248% in Total App Usage Time** and a **0.212% increase in App Usage Time Per User**, with the lower bound of the confidence interval also being the highest among the models tested.

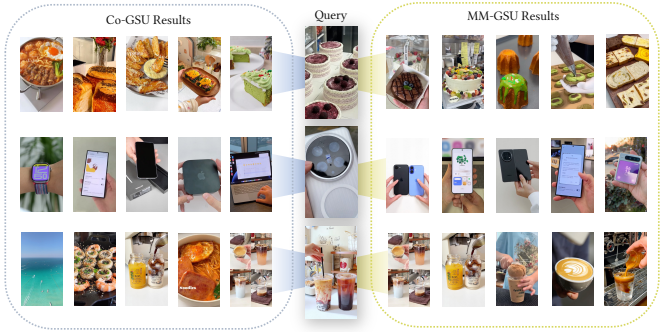

Figure 7: Three specific cases of how searching results can be different between Co-GSU and MM-GSU. In the middle are three query videos. The left column shows searching results of Co-GSU from the user sequence. The right column lists results of MM-GSU.

6.6 Case Study

This section presents three case studies highlighting the different search results of MM-GSU and Co-GSU. We analyze three query videos shown to users and apply both search methods based on their viewing histories, with Top 5 results displayed in Fig. 7.

In the first case, a vlog about a cake baking class yields Co-GSU results that mix general cooking videos, with no specific focus on baking classes. In contrast, MM-GSU's Top 5 results are all centered on cake baking, primarily related to baking classes.

In another case about smartphone fragility, Co-GSU includes various electronics due to co-occurrence learning, which may introduce bias. MM-GSU, however, retrieves semantically relevant smartphone videos.

Lastly, for a coffee-making video, Co-GSU again shows learned co-occurrence, retrieving content beyond coffee, whereas MM-GSU accurately identifies videos specifically about coffee making, demonstrating its contextual relevance.

7 Conclusion

In this paper, we represent the pioneering exploration of leveraging multi-modal information within the advanced tree-based retrieval model. Specifically, for better index structure, we propose multi-modal index tree, which is built using multi-modal embedding to precisely represent video similarity. To precisely capture diverse user interests in user lifelong sequence, we propose Co-GSU and MM-GSU. We conducted a wide range of experiments and our method achieves comparable performance. In the future, we will explore the integration of multi-modal information with other index structure.

References

[1] Xingyan Bin, Jianfei Cui, Wujie Yan, Zhichen Zhao, Xintian Han, Chongyang Yan, Feng Zhang, Xun Zhou, Qi Wu, and Zuotao Liu. 2025. Real-time Indexing for Large-scale Recommendation by Streaming Vector Quantization Retriever. *arXiv preprint arXiv:2501.08695* (2025).

[2] 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 *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2974–2983.

[3] 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 *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3785–3794.

[4] 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).

[5] Rihan Chen, Bin Liu, Han Zhu, Yaoxuan Wang, Qi Li, Buting Ma, Qingbo Hua, Jun Jiang, Yunlong Xu, Hongbo Deng, et al. 2022. Approximate nearest neighbor search under neural similarity metric for large-scale recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3013–3022.

[6] Hao Deng, Haibo Xing, Kanefumi Matsuyama, Moyu Zhang, Jinxin Hu, Hong Wen, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. 2025. CSMF: Cascaded Selective Mask Fine-Tuning for Multi-Objective Embedding-Based Retrieval. *SIGIR 2025* (2025).

[7] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. 2019. Learning to recommend with multiple cascading behaviors. *IEEE transactions on knowledge and data engineering* 33, 6 (2019), 2588–2601.

[8] Weihao Gao, Xiangjun Fan, Chong Wang, Jiankai Sun, Kai Jia, Wenzi Xiao, Ruofan Ding, Xingyan Bin, Hui Yang, and Xiaobing Liu. 2021. Learning an end-to-end structure for retrieval in large-scale recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 524–533.

[9] Zhicheng He, Weiwen Liu, Wei Guo, Jiarui Qin, Yingxue Zhang, Yaochen Hu, and Ruiming Tang. 2023. A survey on user behavior modeling in recommender systems. *arXiv preprint arXiv:2302.11087* (2023).

[10] Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang, and Yong Yu. 2024. A Comprehensive Survey on Retrieval Methods in Recommender Systems. *arXiv preprint arXiv:2407.21022* (2024).

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

[12] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE international conference on data mining (ICDM)*. IEEE, 207–216.

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

[14] Chi Liu, Jiangxia Cao, Rui Huang, Kai Zheng, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. KuaiFormer: Transformer-Based Retrieval at Kuaishou. *arXiv preprint arXiv:2411.10057* (2024).

[15] Qidong Liu, Jiaxi Hu, Yutian Xiao, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Qing Li, and Jiliang Tang. 2024. Multimodal recommender systems: A survey. *Comput. Surveys* 57, 2 (2024), 1–17.

[16] Qijiong Liu, Jieming Zhu, Yanting Yang, Quanyu Dai, Zhaocheng Du, Xiao-Ming Wu, Zhou Zhao, Rui Zhang, and Zhenhua Dong. 2024. Multimodal pretraining, adaptation, and generation for recommendation: A survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6566–6576.

[17] Yifan Liu, Kangning Zhang, Xiangyuan Ren, Yanhua Huang, Jiarui Jin, Yingjie Qin, Ruilong Su, Ruiwen Xu, Yong Yu, and Weinan Zhang. 2024. AlignRec: Aligning and Training in Multimodal Recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1503–1512.

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

[19] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[20] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[21] 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 *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2671–2679.

[22] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

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

[24] Xiang-Rong Sheng, Feifan Yang, Litong Gong, Biao Wang, Zhangming Chan, Yujing Zhang, Yueyao Cheng, Yong-Nan Zhu, Tiezheng Ge, Han Zhu, et al. 2024. Enhancing Taobao Display Advertising with Multimodal Representations: Challenges, Approaches and Insights. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4858–4865.

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

[26] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM web conference 2023*. 845–854.

[27] Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. 2019. Joint optimization of tree-based index and deep model for recommender systems. *Advances in Neural Information Processing Systems* 32 (2019).

[28] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1079–1088.

[29] Jingwei Zhuo, Ziru Xu, Wei Dai, Han Zhu, Han Li, Jian Xu, and Kun Gai. 2020. Learning optimal tree models under beam search. In *International Conference on Machine Learning*. PMLR, 11650–11659.