Mutual Information-aware Knowledge Distillation for Short Video Recommendation

Han Xu[*]
Kuaishou Technology
Beijing, China
xhbj66@gmail.com

Taoxing Pan[*]
Kuaishou Technology
Beijing, China
tx1997@mail.ustc.edu.cn

Zhiqiang Liu
Kuaishou Technology
Beijing, China
zhiqliu1103@gmail.com

Xiaoxiao Xu
Kuaishou Technology
Beijing, China
xuxiaoxiao05@kuaishou.com

Abstract

Short-video sharing platforms engaging billions of users have attracted intense interest recently. A key insight is that user feedback on these platforms is heavily influenced by preceding exposed videos in the same request, called context cumulative effects. For example, multiple repeated videos in a request often cause user fatigue and influence user feedback. However, related factors, such as the other exposed items in the same request, are available during model training but not accessible during online serving. Vanilla distillation methods mitigate the training-inference inconsistency, struggling to capture the dynamic dependence between context cumulative effects and user feedback. To address this problem, we propose the Mutual Information-aware Knowledge Distillation (MIKD) framework, which fuses such effects and user-item matching degrees by evaluating their impacts on user feedback based on mutual information estimation. Rigorous analysis and extensive experiments demonstrate that MIKD precisely extracts personal interests and consistently improves performance. We conduct online A/B testing on a leading short-video sharing mobile app, and the results demonstrate the effectiveness of the proposed method. MIKD has been successfully deployed online to serve the main traffic and optimize user experiences.

CCS Concepts

• Information systems → Recommender systems; **• Computing methodologies → Neural networks**.

Keywords

Context Cumulative Effect,Knowledge Distillation,Mutual Information

ACM Reference Format:
Han Xu, Taoxing Pan, Zhiqiang Liu, and Xiaoxiao Xu. 2025. Mutual Information-aware Knowledge Distillation for Short Video Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3690624.3709403

1 Introduction

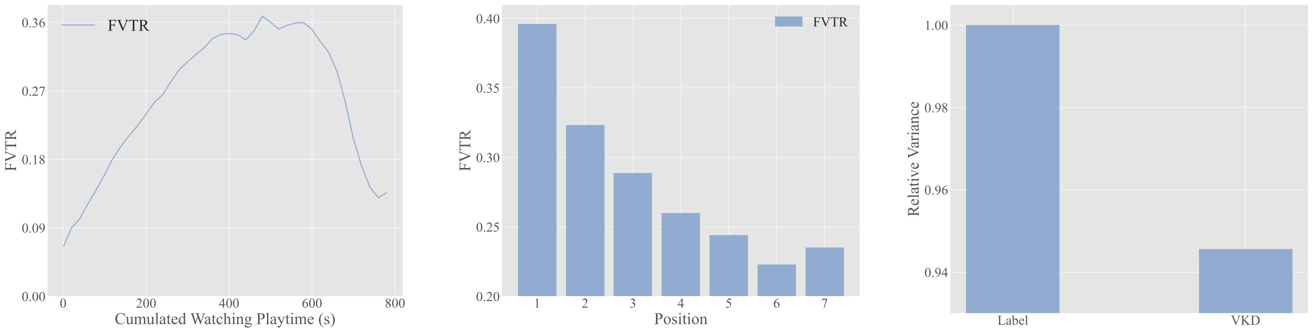
Short-form video-sharing platforms like Douyin, TikTok, YouTube Shorts and Instagram Reels have gained worldwide popularity by delivering interesting videos to their billions of users [4]. Recommending personalized video content is critical for these platforms and has recently attracted intensive interest in academia and industry. From the view of recommender systems, the system returns a set of short videos for users' requests and records their explicit feedback (such as like, follow, collect, etc.) and implicit feedback (such as watching time), where the interactions with the system are based on their preferences. Capturing user preferences from the feedback is a fundamental task for enhancing user engagement and experiences. A key observation is that preceding exposed videos in the same request, besides user interests, can heavily influence user feedback on short video platforms. We call this phenomenon context cumulative effects. For example, Figure 1(a) shows that FVTR (Finish-View-Through Rate, a metric of user watching time) is an unimodal curve of the cumulated watching time for preceding videos in the same request. Other related factors include the position and other exposed videos in the same request. We call these features related to context cumulative effects as request-wise context features.

As request-wise context features only arise after prediction, they are only available in the offline data but unavailable during online serving. To mitigate the training-inference inconsistency, recent works [14, 27] intuitively offer a viable approach, knowledge distillation framework. For example, Xu et al. [27] defines the features not accessible during online serving as post-event features and models the post-event features by vanilla knowledge distillation. While distillation methods effectively address the training-inference inconsistencies, our empirical experiments reveal a noteworthy observation that vanilla distillation methods encounter difficulties in capturing the dynamic user-wise dependencies between context cumulative effects and user feedback. Extensive evidence shows the impact of context cumulative effects on feedback differs among

[*] Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1245-6/25/08
https://doi.org/10.1145/3690624.3709403



(a) FVTR versus the cumulative watching playtime.

(b) FVTR versus exposure position.

(c) The relation between context cumulative effects and predictions

Figure 1: FVTR is a metric about watching time. Figure 1(a) illustrates the relation between the cumulative context playtime and the watching time. Figure 1(b) reveals that the exposure position heavily affects user feedback. Degrees of user feedback influenced by context factors are different among users. Thus, we measure the ability to capture the use-wise dependence between context factors and user feedback by the variance of user-wise correlation between predictions and post-event context features. Figure 1(c) shows that the variance of VKD is smaller than that of the ground truth, indicating that VKD struggles to extract the dynamic dependence between user feedback and post-event contexts.

users. For example, the impact of cumulative watching time on user fatigue varies among individuals.

We propose the Mutual Information-Aware Knowledge Distillation (MIKD) framework to tackle this challenge. MIKD explicitly models the context cumulative effects on user feedback by mutual information, and rigorous analysis shows that MIKD greatly benefits from the explicit modeling module. The teacher model in MIKD consists of mutual information estimation and representation fusion modules. We offer a general method to evaluate the impact of a specific subset of features on the ground-truth labels by mutual information. The teacher learns the embeddings of features related to context cumulative effects and other regular features, respectively, guided by maximizing the mutual information between the representation and the ground-truth distribution. The MIKD fuses the embeddings based on the mutual information evaluation. Statistic analysis demonstrates that our proposed method makes improvements in capturing the dynamic dependence between user feedback and context cumulative effects.

We conduct extensive experiments to validate the effectiveness and generality of MIKD. Our experiments are conducted in the real-world settings of a leading short video platforms with millions of daily active users [10]. Note that MIKD requires a set of exposed items in the same request for training, but traditional short video datasets lack such contextual information. Thus, we collect a production dataset comprising millions of instances from the online data logs. To promote further research, the anonymized dataset and the code are available on GitHub at https://github.com/taoxingpan/MIKD. Besides, we conduct offline experiments on existing public datasets, Avito dataset, to validate the generality of our methods. The experiments indicate that MIKD consistently and significantly enhances the prediction performance of all baseline models, and outperforms the vanilla knowledge distillation framework. Furthermore, we test our proposed MIKD algorithm through online experiments and observe a 0.680% improvement in watching time (WT) without introducing additional latency, while a 0.1% increase in Watch Time is a practical improvement. Our methods have been successfully deployed online for half a year, serving the main traffic and optimizing user experience. In summary, our main contributions are as follows:

- We present a key observation for short-form video-share platforms that context cumulative effects heavily influence user feedback. To the best of our knowledge, we are the first to study the context cumulative effect in short-video sharing platforms.
- From detailed analysis, we observe that the context cumulative effect on user feedback varies among users and that vanilla knowledge distillation has difficulty in capturing such user-wise dynamic dependence. To address this problem, we propose the Mutual Information-aware Knowledge Distillation (MIKD) framework, which fuses context cumulative effects and user-item matching degrees by evaluating their impacts on user feedback based on mutual information estimation. MIKD has been successfully deployed online to serve the main traffic.
- We will release a large-scale industrial dataset to facilitate further research in this novel direction upon acceptance.

2 Related Work

2.1 Post-event Features Modeling

Recent research in traditional recommender systems, such as advertising and e-commerce [29], leverages post-event features to enhance the precision of predictions. Research [1, 15, 21, 24, 25] investigated the direct modeling of the positions where items are displayed and their impact on user engagement. Techniques like outcome randomization [15, 21, 24] and inverse propensity weighting (IPW) [1, 25] were employed to mitigate the impacts associated with the exposed positions. One approach, known as the Position-Aware Learning (PAL) framework [13], models exposure positions by assuming that the probability of an item being clicked depends on the sum of the probability of the item being seen and the probability of the user clicking the item. These two modules are jointly optimized simultaneously. Additionally, the post-click behavior of users has been recognized as valuable for predicting subsequent actions, such as making a purchase. Xu et al. [27], Yang et al. [31] proposed Privileged Feature Distillation (PFD), modeling user clicks to make prediction purchase more precisely.

2.2 Short Video Recommendation

Short Video recommendation is an important application of recommender models. At the heart of short video recommendation is accurately capturing user interests based on their explicit and implicit feedback with exposed videos. Recent research efforts, such as those by Quan et al. [20], Zhan et al. [32], Tang et al. [22], Xu et al. [28] and Lin et al. [19], honed in on the critical aspect of user watching time to precisely preference extraction. Lin et al. [19] introduced a novel framework that predicts the expected watching time by traversing tree structure. Quan et al. [20], Zhan et al. [32] and Tang et al. [22] collectively highlighted the dual influences on user feedback: the content's appeal and the inherent duration of the video, noting that long-duration videos inherently encourage extended watching time. Zhan et al. [32] and Tang et al. [22] applied causal inference techniques to mitigate the confounding effects of video duration on user satisfaction. In addition, Quan et al. [20] developed a data labeling strategy and a sample generation process that effectively discerns user preferences from the perspective of watching time. Expanding on these findings, we suggest that the user's immediate viewing history within the same request also influences user feedback. Our research introduces a novel approach that integrates the sequential influence of a user's watching history into our recommendation model to enhance the precision of interest modeling.

3 Background And Motivation

The short video recommendation system interacts with users in a streaming manner. When the recommender receives a request, it returns a list of videos in real-time. Unlike traditional systems where a single screen displays all items, users view the next video within the same request by scrolling up in a streaming manner, as illustrated in Figure 2. The video with the next exposure position is presented when the user scrolls up the screen.

The current video's exposed position and the context's accumulated playtime influence user feedback for short videos. Among all kinds of user interactions, the main goal of the short video delivery is to optimize the watching time, as watching time reflects user attention and preferences. We define Finish-View-Through Rate (FVTR), which equals 1 if the watching time reaches the 75th percentile watching time and 0 otherwise. This signal positively correlates with watching time in short-video sharing platforms. We observe that FVTR sharply decreases as the presentation position increases, shown in Figure 1(b). Simultaneously, FVTR is also influenced by the cumulative playtime under the same request, shown in Figure 1(a). Interestingly, FVTR is not monotonic with respect to the accumulated context playtime. Thus, ignoring the impact of context cumulative effects may harm prediction accuracy and adversely affect the user experience. Consequently, modeling context cumulative effects in short video platforms is important and necessary, as it is a significant factor for prediction accuracy. Since request-wise context features take effect after prediction, related features cannot be available during online serving. Inspired by other works [27], we employ vanilla knowledge distillation (VKD) to model information in the same request to address consistency between serving and offline training.

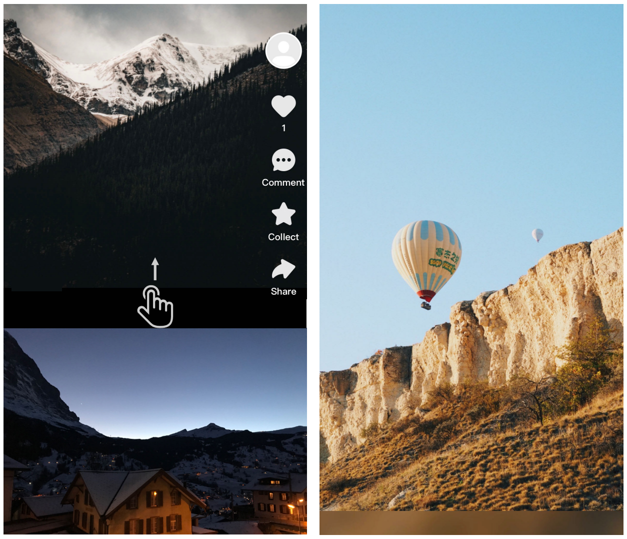


Figure 2: An example of a popular short video platform.

While the vanilla distillation improves performance, we observe that the method has difficulty extracting the user-wise context cumulative effects on user feedback. To describe the problem, we introduce Spearman's rank correlation r_s, which assesses monotonic relationships between two variables X, Y [26]. Given a dataset $\{(x, y)_i^n\}$, the Spearman's rank correlation is computed as

$$r_s = \frac{cov(R_X, R_Y)}{\sigma_{R_X} \sigma_{R_Y}}, \tag{1}$$

where R_X, R_Y are the ranks of the corresponding variables x, y, $cov(\cdot, \cdot)$ is the covariance, and σ is the standard variance. For simplicity, we denote a user as $u \in U$ and the dataset generated by the user as $\mathcal{D}_u$. Following the definition, we measure the user-wise relation between context cumulative effects and predictions by Spearman's rank correlation r_s^u, which is computed over a given user's dataset $\mathcal{D}_u = \{(x_i^u, y_i^u)\}$. Then, we measure the dispersion of such user-wise relation by variance, that is,

$$\mathbb{V}\left(r_s\left(R_{X^u}, R_{Y^u}\right)\right), r_s \in \{r_s^u : u \in U\}, \tag{2}$$

and denote the variance by $v_U(X, Y) = \mathbb{V}_U(r_s(R_{X^u}, R_{Y^u}))$. We describe the dynamic dependence between request-wise context features X_c and user feedback Y by the dispersion of Spearman's rank correlation, i.e., $v_U(X_c, Y)$. As for the prediction $\hat{Y}$, we have

$v_U(X_c, \hat{Y})$. Figure 1(c) shows that $v_U(X_c, Y)$ is larger than $v_U(X_c, \hat{Y})$, which indicates that VKD has difficulty in extracting the dynamic dependence among users. Therefore, we propose a method to model the user-wise context cumulative effects on the feedback and combine the method with the knowledge distillation framework.

4 Mutual Information Knowledge Distillation Framework

To tackle the abovementioned problem, we propose the Mutual Information-aware Knowledge Distillation (MIKD) framework to distinguish context cumulative effects and user preferences on user feedback by mutual information estimation. Based on the estimation, the teacher model in MIKD fuses all regular features and the features related to context cumulative effects and guides the learning process of the student model. In this section, we first introduce the Mutual Information Estimation method and then show the training procedure and online deployment of Mutual Information-aware Knowledge Distillation (MIKD) framework.

4.1 Mutual Information Estimation

For simplicity, this paper presents the formal definition in Table 1. Given a dataset $\mathcal{D}$, and a model $\hat{y} = f_\theta(h_w(\cdot))$, the optimization objective is that

$$\min_{w,\theta} d(\hat{Y}, Y) = \frac{1}{n} d(f_\theta(h_w(\mathbf{x}_i)), y_i). \tag{3}$$

In general, the impact of different input features on the ground-truth labels Y is different. For example, the user's feedback on a specific item depends on quantity factors in a real-world recommendation system. We describe the importance of a given subset of feature $A \subset \{1, 2, \ldots, m\}$ as the amount of information of labels that the subset A contains. Denote the selected feature space as $\mathcal{X}_A = \Pi_{i \in A} \mathcal{X}_i$. The proper scoring rule to evaluate the importance of a subset features is a function over a set $I : A \longrightarrow [0, 1]$, where the universe is $\Omega = [m] = \{1, 2, \cdots, m\}$, and the function should satisfy the following properties at least:

1. (1) The importance of the empty feature set equals to 0, i.e., $I(\emptyset) = 0$.
2. (2) The importance of the universe equals to 1, i.e., $I(\Omega) = I([m]) = 1$.
3. (3) Given two subset A, B of Ω, if $I(A) \geq I(B)$, then there exists an embedding function h_{w^*} such that $I(h_{w^*}(X_A); Y) \geq \sup \{I(h_w(X_B); Y) : h_w\}$.

Given a subset of features A, if $A \notin \{\emptyset, \Omega\}$, the selected features $\mathbf{x}_A \in X_A$ is a random variable with probability distribution $X_A \sim p_{X|X_{\Omega \setminus A}}$. We show that the normalized mutual information

$$I(A) = \begin{cases} \dfrac{I(X_A; Y)}{I(X; Y)} & \text{if } A \notin \{\emptyset, \Omega\}, \\ 0 & \text{if } A = \emptyset, \\ 1 & \text{if } A = \Omega, \end{cases} \tag{4}$$

is a proper scoring rule. The proof of properties (1) and (2) is simple. Obviously, if one of the subset A, B is the empty set or the universe, property (3) establishes. Since the case above is trivial, we only focus on the case $A \in 2^\Omega \setminus \{\emptyset, \Omega\}$ in the remaining paper. Before establishing property (3), we introduce the universal approximation theorem, which Funahashi [9] provides a rigorous proof for arbitrary width neural networks.

Theorem 4.1. *Let $C(S, \mathbb{R}^m)$ denote the set of continuous function from a compact subset S of a Euclidean $\mathbb{R}^n$ space to a Euclidean space $\mathbb{R}^m$. Given an output function $\phi(x), x \in \mathbb{R}^n$, for any $h \in C(S, \mathbb{R}^m)$, an arbitrary $\epsilon > 0$, there exists N, and the network parameters $c_i, b_i, (i = 1, 2, \cdots, N), w_{ij}, (i = 1, 2, \cdots, N, j = 1, 2, \cdots, n)$ such that*

$$\sup_{\mathbf{x} \in S} \|h(\mathbf{x}) - g(\mathbf{x})\| < \epsilon, \tag{5}$$

where

$$g(x) = \sum_i^N c_i \phi \left(\sum_j^n w_{ij} x_j + b_i \right). \tag{6}$$

Theorem 4.1 shows that the neural network is able to approximate any continuous functions of Euclidean space. Thus, there exists an embedding function h_{w^*} like space-filling curves such that

$$I(h_{w^*}(X_A); Y) \geq \sup \{I(h_w(X_B); Y) : h_w\},$$

if $I(X_A) \geq I(X_B)$. That is, the normalized mutual information satisfies property (3). Since the mutual information $I(X_A; Y)$ is proportional to $I(A)$, we measure the importance of a subset of features by the mutual information. Note that the cross entropy between the ground-truth labels and predictions $\mathcal{H}(Y; \hat{Y})$, is inversely proportional to the mutual information [3].

Theorem 4.2. *The optimal embedding layer h_{w^*} simultaneously maximizes the mutual information $I(\hat{Z}; Y)$ and minimizes the cross-entropy $\mathcal{H}(Y; \hat{Y})$.*

The detailed proof of Theorem 4.2 is provided in the paper [3]. Based on Theorem 4.2, we use the standard cross-entropy to approximate the importance of a subset features:

$$I(A) := I(X_A; Y) \approx \frac{1}{\mathcal{H}(Y; \hat{Y})}. \tag{7}$$

Furthermore, in real-world recommendation systems, the impact of different features on user feedback differs among users. For example, multiple repeated videos in the same request cause different degrees of user fatigue among users. This phenomenon motivates us to model the dynamic dependence between the selected features and the ground-truth labels. Given a specific user u_i, we denote the samples of this user as $(X, Y|u_i) \sim p_{\mathbf{x},y|u_i}$. Following the property of the mutual information and the cross entropy over the samples $(X, Y|u_i)$, we have

$$\begin{aligned} I(A|u_i) &:= I(X_A; Y|u_i) \\ &\approx \frac{1}{\mathcal{H}(Y; \hat{Y}|u_i)} = \frac{1}{\mathbb{E}_{p_{\mathbf{x},y|u_i}} \left[-p_{\mathbf{x},y|u_i} \log \hat{p}_{\mathbf{x},y|u_i} \right]}. \end{aligned} \tag{8}$$

4.2 Context Cumulative Effect Awareness Modeling via MIKD

Based on the importance of a subset features, we propose MIKD to model the dynamic dependence between post-event features and the ground-truth labels. MIKD splits the entire features into two

Table 1: Definitions of the sample data, selected features and information measures used in the paper.

Dataset and Features

Sample Dataset	$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
Single feature space	$\mathcal{X}_j$
Total features space	$\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$
Label/Prediction space	$\mathcal{Y} \subset \mathbb{R}$
Embedding space	$\mathcal{Z} \subset \mathbb{R}^d$
Features Encoder	$h_w : \mathcal{X} \to \mathcal{Z}$
Output Layer	$f_\theta : \mathcal{Z} \to \mathbb{R}$

Random Variables

Observation/Features random variable	X
Label random variabl	Y
Estimation random variabl	$\hat{Y}$
Observation distribution	$X \sim p_X$
Label distribution	$Y\|X \sim p_{Y\|X}(\cdot\|X)$
Embedding	$\hat{Z}\|X = h_w(X)$
Prediction	$\hat{Y}\|\hat{Z} = f_\theta(\hat{Z})$

Information Measures

Entropy of the ground-truth labels Y	$\mathcal{H}(Y) := \mathbb{E}_{p_Y}[-\log p_Y(Y)]$
Conditional entropy of Y given the embedding $\hat{Z}$	$\mathcal{H}(Y\|\hat{Z}) := \mathbb{E}_{p_{Y\|\hat{Z}}}[-\log p_{Y\|\hat{Z}}(Y\|\hat{Z})]$
Cross entropy (CE) between Y and $\hat{Y}$	$\mathcal{H}(Y;\hat{Y}) := \mathbb{E}_{p_Y}[-\log p_{\hat{Y}}(Y)]$
Mutual information between $\hat{Z}$ and Y	$\mathcal{I}(\hat{Z};Y) := \mathcal{H}(Y) - \mathcal{H}(Y\|\hat{Z})$
Probability distance between $\hat{Z}$ and Y	$d(\hat{Z}, Y)$

parts $\Omega = A_r \cup A_c$, where A_r contains regular features and A_c contains context cumulative effects related features. The corresponding random variables are denoted by X_r, X_c. As the request-wise context features are available during training but not accessible during online serving, MIKD conducts distillation to alleviate training-inference inconsistency. In MIKD, the input of the teacher model for training is all features X_r, X_c, while we only use regular features X_r for the student model. The teacher model consists of two fully separated towers to learn the representation of regular features and request-wise context features, respectively. We denote the embedding layer and the output layer for regular features as $h_w^r(\cdot)$ and f_θ^r, while the other embedding layer and output layer are denoted by $h_w^c(\cdot)$ and f_θ^c. MIKD trains these two towers with cross entropy loss on the user's dataset $\mathcal{D}_u$, respectively, and the losses are denoted by l_r and l_c. Since theorem 4.2 illustrates that minimizing cross entropy $\mathcal{H}(Y;\hat{Y}|u_i)$ is equivalent to maximizing the mutual information between $\hat{Z}$ and Y, we also call them the **M**aximized mutual **I**nformation loss, $\mathcal{L}_{MI}$,

$$\mathcal{L}_{MI} := \mathbb{E}_{p_{\mathbf{x},y|u_i}}\left[-p_{\mathbf{x},y|u_i}\log\left(\hat{p}_{\mathbf{x},y|u_i}\right)\right] \tag{9}$$

$$= \sum_{(\mathbf{x},y)\in\mathcal{D}_{u_i}} \frac{-y\log\hat{y}}{|\mathcal{D}_{u_i}|} = \sum_{(\mathbf{x},y)\in\mathcal{D}_{u_i}} \frac{-y\log f_\theta(h_w(\mathbf{x}))}{|\mathcal{D}_{u_i}|}, \tag{10}$$

where $|A|$ is the number of elements in the set A. Note that MIKD needs group samples by user in a batch in the training for $\mathcal{L}_{MI}$. The teacher model makes a weighted sum on the embedding based on the user-wise Maximized mutual Information loss, i.e.,

$$\alpha_r = \frac{(l_r)^{-1}}{(l_r)^{-1} + (l_c)^{-1}}, \quad \alpha_c = \frac{(l_c)^{-1}}{(l_r)^{-1} + (l_c)^{-1}},$$
$$\hat{z}_{\mathbf{x}} = \alpha_r h_w^r(\mathbf{x}_r) + \alpha_c h_w^c(\mathbf{x}_c). \tag{11}$$

Then, the fusion layer $f_\theta^t : \mathcal{Z} \to \mathbb{R}$ in the teacher outputs the prediction $\hat{y}^t$ based on $\hat{z}_{\mathbf{x}}$. In this paper, the fusion module is an multilayer perceptron. MIKD uses the output to guide the training of the student model by the cross entropy loss,

$$\mathcal{L}_{teacher} = -\frac{1}{n}\sum_{i=0}^{n} y\log(\hat{y}^t), \tag{12}$$

$$\mathcal{L}_{distillation} = -\frac{1}{n}\sum_{i=0}^{n} \hat{y}^t\log(\hat{y}^s), \tag{13}$$

where the teacher output $\hat{y}^t = f_\theta^t(\hat{z}_{\mathbf{x}})$ is the soft label. The distillation loss is beneficial for the student to model the context cumulative effects on user feedback. Notice that we freeze the parameters in the teacher model in training of $\mathcal{L}_{distillation}$. Besides, the student optimizes the probability distance between the prediction and the ground-truth distribution by minimizing the cross entropy loss, that is, $\mathcal{L}_{student}$. The total loss in MIKD is

$$\mathcal{L} = \mathcal{L}_{teacher} + \mathcal{L}_{student} + \alpha\mathcal{L}_{distillation} + \mathcal{L}_{MI}^r + \mathcal{L}_{MI}^c, \tag{14}$$

where α is a hyperparameter.

In conclusion, given a dataset consists of two kinds of features, regular features X_r and post-event features X_c, two towers in the teacher model learn the probability distribution $P(Y|X_r)$ and $P(Y|X_c)$ by the maximized mutual information loss $\mathcal{L}_{MI}^r$ and $\mathcal{L}_{MI}^c$ respectively. The teacher model fuses these two encoder embedings by mutual information metric and outputs the final teacher prediction $\hat{y}^t$, which serves as a soft label to guide the learning process of the student model.

For online serving, we exclusively deploy the student model to improve the user experience. The knowledge about context cumulative effects from the teacher model is transferred to the student via the distillation process. Notice that during serving, only regular features X_r are used in the student model, and neither the post-event context nor the teacher model are involved. As a result, this approach ensures that latency remains consistent with the baseline.

5 Experiments

In this section, we present and analyze extensive experiments to show the effectiveness of our proposed method in detail. We introduce the real-world industrial dataset and the public dataset in subsection 5.1. We explain the choice of metrics, backbones, and advanced techniques for comparison in subsections 5.2 and 5.3. We present the performance comparison results in subsection 5.4 and ablation studies in subsection 5.5, respectively. The experiments illustrate the effectiveness and generality of our methods. We analyze the appealing properties induced by MIKD in subsection 5.6. Finally, we give the results of online A/B Testing in subsection 5.7.

5.1 Dataset

Public available datasets of short-form video-sharing platforms, such as KuaiRand [10], lack the log of other exposure videos on the same page. However, user feedback is heavily influenced by other exposed videos, a phenomenon we refer to as the context cumulative effect. To support further academic research on this effect, we collect one real-world industrial dataset, Video-Expo, from the online logs and make the dataset public. Our primary offline experiments and ablation studies are conducted on Video-Expo, and we also present the corresponding online A/B testing results. Furthermore, we conduct experiments on Avito context dataset [1] to validate the generality of our methods.

5.1.1 Video-Expo. Due to the limitation of available public datasets, we build a dataset from online logs of the recommendation system. We randomly sample a subset of the users, and use their online logs from 2023-10-05 to 2023-10-16, a total of 12 days, for training and the instances on the next day for testing. Every instance in the dataset consists of user features, video features, request features, and the label. User features include user ID, age ID, and gender ID. Video features consist of video ID, author ID, category ID, duration, etc. Each request is associated with the corresponding <request ID, video ID, user ID, request timestamp, position, a list of other exposed video IDs, and the accumulated watch time on the request>. The label is FVTR, short for Finish-View-Through Rate, which equals 1 if the watch time reaches the 75th percentile watch time and 0 otherwise. All items, users, and request features are represented with unique and anonymous IDs for privacy protection. Statistics are shown in Table 3, where the avg. length is the average numbers of items on the same page.

5.1.2 Avito-Exposure. Avito dataset consists of random samples of logs from Avito.ru, which is Russia's largest classified advertisements website. This dataset contains features of users, searches, and advertisements, including user_id, search_query, search_id, ad_title, ad_category, and other features. As the probability of click events is influenced by other advertisements on the same search page and the corresponding exposure positions, we consider these two features as request-wise context features. The dataset has been widely used as a benchmark for CTR Prediction tasks. Following recent works [8, 18], the training data consists of the ad logs from 2015-04-28 to 2015-05-18 for training, those on 2015-05-19 for validation, and those on 2015-05-20 for testing. Statistics about Avito dataset are shown in Table 3.

Table 2: Evaluation results on Avito-Expo and Video-Expo datasets. Vanilla Knowledge Distillation and MIKD are integrated with three backbone networks, DNN, DCN and DeepFM. The results are averaged over 5 runs. Std ≤ 0.1%.

Backbone	VKD	MIKD	Video-Expo		Avito	
			AUC	GAUC	AUC	GAUC
DNN			0.7936	0.7544	0.7737	0.7158
DNN	✓		0.7937	0.7540	0.7763	0.7186
DNN		✓	0.7949	0.7551	0.7784	0.7188
DCN			0.7911	0.7515	0.7745	0.7169
DCN	✓		0.7937	0.7539	0.7755	0.7174
DCN		✓	0.7944	0.7542	0.7782	0.7196
DeepFM			0.7929	0.7536	0.7759	0.7174
DeepFM	✓		0.7941	0.7543	0.7776	0.7195
DeepFM		✓	**0.7959**	**0.7564**	**0.7827**	**0.7212**

Table 3: Statistics of datasets.

Dataset	#user	#item	avg. length	#Instance
Video-Expo	0.40M	36M	6.4	720M
Avito	2.4M	0.028M	3.3	71M

5.2 Metrics

We adopt AUC as the evaluation metric. AUC represents the probability that a positive sample's score is higher than a negative one, reflecting a model's ranking ability [2, 30, 34, 35]. In addition, we group instances by their user IDs and use GAUC [34] (Group AUC) to evaluate the overall performance of local pairwise accuracy in each group, i.e.,$\text{GAUC} = \frac{\sum_i^u \omega_i \text{AUC}_i}{\sum_i^u \omega_i}$, where u is the number of users, ω_i is the number of samples of i-th user, and AUC_i is the intra-user AUC. Both AUC and GAUC are consistent with online performance. For a fair comparison, each model is repeatedly trained and tested 5 times on both industry and public datasets, and the average results are reported.

5.3 Backbones & Settings

Since our proposed MIKD framework is plug-and-play, we conduct our experiments on three classical baselines: DNN [6], DCN [23], and DeepFM [12]. The detailed network architectures are as follows:

- DNN follows an Embedding&MLP paradigm, which applies MLP for high-order feature interaction. It is worth noting that DNN serves as the base of most CTR prediction models.
- DCN explicitly applies feature crossing at each layer, and the deep part is the same as DNN.
- DeepFM combines the power of factorization machines for recommendation and deep learning. The deep part is the same as DNN.

[1] https://www.kaggle.com/c/avito-context-ad-clicks/data

As request-wise context features are similar to post-event features, we choose several state-of-the-art post-event feature-aware click prediction models for comparison. Besides, we introduce the commonly used feature selection method. That is,

- PAL [13] is a bias-aware model which assumes the click probability depends on the probability that the item is seen by the user and the user preference. In this work, PAL is applied to the backbone structure.
- Dropout [33] takes the post-context information as a feature and applies the dropout trick. The dropout rate here is 0.1, consistent with [33].
- SeNet [5] explicitly models the interdependence between the channels of convolutional features in the image classification tasks. Following the prevent work [17], we use the SeNet mechanism to learn the weights of the regular features and request-wise context features for the fusion in the teacher model.
- Deep Position-wise Interaction Network (DPIN) [16] captures deep non-linear interactions among position, user, context, and item to estimate the CTR at each position. Assuming an equal probability of item appearance across all positions, we aggregate predictions of DPIN across positions to derive its final output.
- CLID [11] is the state-of-the-art distillation method to model post-event context features for CTR prediction, which employs a listwise distillation task to consider other items on the same page.

5.3.1 Settings. In the experiments, a three-layer MLP with 256, 128, and 128 hidden units serves as the main tower for all backbones. We use AdaGrad [7] optimizer. The hyper-parameters are set as follows: batch size = 1024 and learning rate = 0.001. As for the hyperparameter α of distillation loss, we set $\alpha = 2.0$ for all knowledge distillation methods.

5.4 Performance Comparison

5.4.1 The results on industrial dataset. The results presented in Table 2 illustrate that MIKD achieves a significant and consistent performance gain across all baselines. Compared to vanilla distillation, our approach achieves increases of up to 0.18% in AUC and 0.21% in GAUC, which are considerable gains as even a 0.1% improvement in AUC is substantial for industrial systems. These results indicate a marked enhancement in ranking performance induced by MIKD. Besides, we provide standard deviations in Table 7. The results show the reliability of our methods.

5.4.2 Results on public dataset. To evaluate the general applicability of our framework, we conduct offline experiments on Avito dataset. As depicted in Table 2, our method consistently outperforms vanilla distillation across various backbones, achieving gains in both AUC and GAUC. These outcomes further confirm the effectiveness and generality of our method.

5.4.3 Comparison of post-event features modeling and feature selection methods. We compare several state-of-the-art post-event feature-aware models and feature selection methods. Specifically, due to its succinct implementation, we carefully tune PAL, Dropout, DPIN and CLID with DeepFM. As shown in Table 4, while other methods improve the AUC and GAUC metrics, their performance is still inferior to that of our method. Notice that CLID [11] is the state-of-the-art distillation method for post-event feature modeling.

Table 4: Performance comparison to post-event features modeling and feature selection methods on an industrial dataset

Datasets	Methods	AUC	GAUC
Video-Expo	DeepFM+PAL	0.7932	0.7547
	DeepFM+Dropout	0.7923	0.7530
	DeepFM+VKD+SeNet	0.7936	0.7541
	DeepFM+DPIN	0.7938	0.7537
	DeepFM+CLID	0.7951	0.7553
	Ours	**0.7959**	**0.7564**

Table 5: Ablation results of several key components.

Model	Teacher		Student	
	AUC	GAUC	AUC	GAUC
logits-based (ours)	0.8048	0.7648	0.7959	0.7564
feature-based	0.8027	0.7631	0.7928	0.7532
w/o KL	0.8019	0.7623	0.7928	0.7539
w/o $Loss_{MI}$	0.8023	0.7616	0.7922	0.7526
Gate network	0.8033	0.7628	0.7953	0.7539
w/o MI-Fusion	0.8029	0.7626	0.7945	0.7549

Table 6: Online A/B test. In real world's scenario, 0.1% increase at Watch Time is a significant, which brings great business effectiveness.

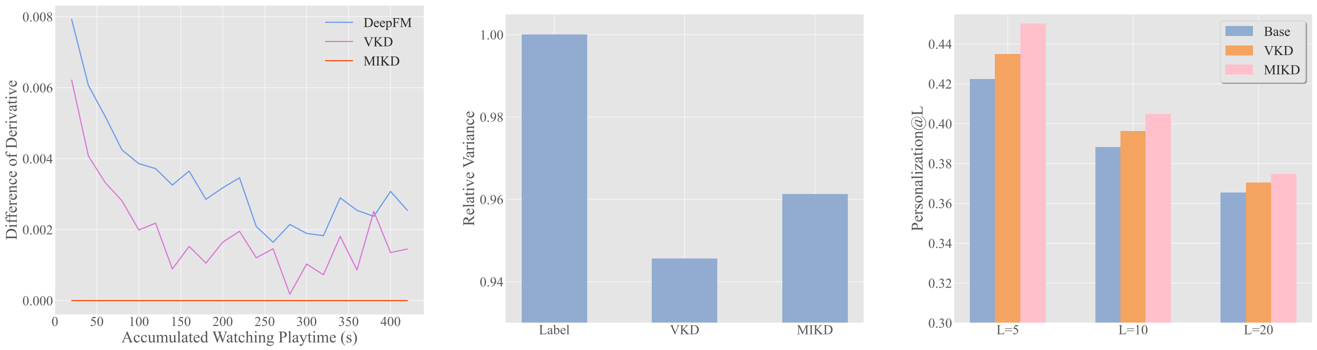
Method	WT	CI	Latency
VKD	+0.219%	[0.115%,0.322%]	± 0.00ms
MIKD	+0.680%	[0.575%,0.785%]	± 0.00ms

In addition to comparing with standard post-event features modeling methods, we compare SeNet[5], a method for feature selection, to our method MIKD. Based on the vanilla distillation framework, we implement SeNet to model the importance of regular features and request-wise context features at the feature embedding layer. Then, the features, weighted by their importance, are fused and fed into the teacher model. As indicated in Table 4, the results show that MIKD outperforms all methods mentioned above.

5.5 Ablation Study

To illustrate the contribution of each component in MIKD, we conduct ablation studies on the industrial dataset, Video-Expo.

5.5.1 The effect of knowledge distillation. We conduct the offline experiments to assess the impact of knowledge distillation. Specifically, we remove the logit distillation loss during the training stage. The results in Table 5 indicate that the absence of distillation loss leads to decreases in both AUC and GAUC for both the teacher and student models, demonstrating the significance of knowledge distillation framework.



(a) Sensitivity of request-wise context features

(b) Normalized $\nu_U(X_c, Y)$ and $\nu_U(X_c, \hat{Y})$

(c) Results of Personalization

Figure 3: We calculate the derivatives of predictions with respect to the accumulated context playtime, denoted by $\nabla \hat{y}_{DeepFM}$, $\nabla \hat{y}_{VKD}$, $\nabla \hat{y}_{MIKD}$. Because of the minor differences among them, we present the discrepancy $(\nabla \hat{y} - \nabla \hat{y}_{MIKD})$ in Figure 3(a). The figure indicates that $\nabla \hat{y}_{DeepFM}$ and $\nabla \hat{y}_{VKD}$ exceed $\nabla \hat{y}_{MIKD}$, signifying that MIKD has a lower sensitivity in comparison to the other two methods. Similar to figure 1(c), figure 3(b) presents the normalized $\nu_U(X_c, Y)$, $\nu_U(X_c, \hat{Y}_{VKD})$ and $\nu_U(X_c, \hat{Y}_{MIKD})$. The figure demonstrate that MIKD outperforms VKD in capturing the dynamic dependence between request-wise context features and labels. Figure 3(c) compares the personalization@L among three methods when setting $L = 5, 10, 20$, indicating that MIKD achieves superior performance in personalization.

5.5.2 The effect of maximizing mutual information loss. We remove the maximum mutual information loss from MIKD to assess the effectiveness of the loss. The outcomes shown in Table 5 demonstrate that the absence of the loss weakens the performance.

5.5.3 Necessity of the mutual information-aware fusion. We initially remove the mutual information-aware fusion module and concatenate the embeddings as the input of the student. Subsequently, we observe a decline in AUC decreases to 80.29% and 79.45% for the teacher and student models. Besides, GAUC decreases to 76.26% and 75.49%. Table 5 shows the detailed results. In addition, replacing our mutual information-aware fusion with a gate network leads to inferior performance.

5.6 Qualitative Results

To gain a comprehensive understanding of experimental results and further verify their effectiveness, we analyze the offline outputs and recommendation results generated by baseline, VKD, and MIKD. The first analysis is to calculate the derivatives of predictions w.r.t. cumulative context playtime, denoted as $\nabla \hat{y}_{DeepFM}$, $\nabla \hat{y}_{VKD}$, $\nabla \hat{y}_{MIKD}$. Figure 3(a) shows both the derivatives of the backbond and vanilla knowledge distillation exceed that of MIKD, indicating that MIKD is less sensitive to the accumulated context playtime than other methods. The analysis shows that MIKD captures genuine user preferences by mitigating the impact of context cumulative effects on predictions. Subsequently, we compare the ability to model the different context cumulative effects among users, as outlined in Equation 2. Figure 3(b) illustrates that MIKD achieves the best performance on capturing the user-wise dynamic dependence between the user feedback and context cumulative effects. Finally, we compare Personalization@L [12, 13] metric among MIKD, VKD, and the baseline. Personalization@L can measure the inter-user diversity of top-L items in a ranking list across different users, reflecting the degree of personalization recommendations provided. Personalization@L is defined as:

$$h(a,b) = 1 - \frac{q_{ab}(L)}{L}, \quad a, b \in U, \tag{15}$$

$$Personalization@L = \frac{1}{|U| \times (|U|-1)} \sum_{a \in U} \sum_{b \in U} h(a,b), \tag{16}$$

where $|U|$ is the size of the user group U, $q_{ab}(L)$ represents the quantity of common categories within the top-L recommended items shared between users a and b. We calculate Personalization@L from online logs when setting $L = 5, 10, 20$ respectively. Figure 3(c) indicates that recommended items generated by MIKD are more diverse than those generated by other models, demonstrating that MIKD yields the highest level of personalization.

5.7 Online A/B Testing

We conduct rigorous online A/B tests on a billion-user scale short-video platform. Table 6 presents a performance comparison of MIKD against KD and the industrial baseline. Unlike e-commerce, where online evaluation metrics commonly include CTR and GMV, short-video recommendation scenarios prioritize Watch Time. As illustrated in Table 6, MIKD significantly outperforms KD and the baseline models, with no additional latency at the serving stage. In detail, MIKD achieves a 0.68% and a 0.46% improvement in watching time compared with the base model and vanilla knowledge distillation. As a 0.1% increase in Watch Time is a practical improvement, MIKD achieves significant business gain. Besids, we achieve a 0.608% increase in the number of effective view. MIKD is deployed in our online service, serving millions of daily users and improving their experiences.

6 Conclusions

Short-form video-sharing platforms have achieved tremendous success in business by delivering interesting short videos to billions of users. The recommender systems of short video platforms return a set of short videos for users' requests and record both explicit and implicit feedback. In this paper, we observe that context cumulative effects can heavily influence user feedback. To model such effects, we propose the Mutual Information-aware Knowledge Distillation (MIKD) framework, which fuses request-wise context features and other regular features by evaluating their impacts on user feedback based on mutual information estimation. Extensive experiments demonstrate that our proposed method captures the user-wise dynamic dependence between user feedback and context cumulative effects and consistently improves performance. Besides, MIKD has been successfully deployed online to serve the main traffic and optimize user experiences.

References

[1] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st international ACM SIGIR conference on research & development in information retrieval.* 385–394.

[2] Vincent J Aidala. 1979. Kalman filter behavior in bearings-only tracking applications. *IEEE Trans. Aerospace Electron. Systems* 1 (1979), 29–39.

[3] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. 2020. A Unifying Mutual Information View of Metric Learning: Cross-Entropy vs. Pairwise Losses. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI.* 548–564.

[4] Qingpeng Cai, Zhenghai Xue, Chi Zhang, Wanqi Xue, Shuchang Liu, Ruohan Zhan, Xueliang Wang, Tianyou Zuo, Wentao Xie, Dong Zheng, et al. 2023. Two-Stage Constrained Actor-Critic for Short Video Recommendation. In *Proceedings of the ACM Web Conference 2023.* 865–875.

[5] Dongcai Cheng, Gaofeng Meng, Guangliang Cheng, and Chunhong Pan. 2016. SeNet: Structured edge network for sea–land segmentation. *IEEE Geoscience and Remote Sensing Letters* 14, 2 (2016), 247–251.

[6] 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.

[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* (2011), 2121–2159.

[8] Zhifang Fan, Dan Ou, Yulong Gu, Bairan Fu, Xiang Li, Wentian Bao, Xin-Yu Dai, Xiaoyi Zeng, Tao Zhuang, and Qingwen Liu. 2022. Modeling Users' Contextualized Page-Wise Feedback for Click-Through Rate Prediction in E-Commerce Search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining.* 262–270.

[9] Ken-Ichi Funahashi. 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2, 3 (1989), 183–192.

[10] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management.* 3953–3957.

[11] Xiaoqiang Gui, Yueyao Cheng, Xiang-Rong Sheng, Yunfeng Zhao, Guoxian Yu, Shuguang Han, Yuning Jiang, Jian Xu, and Bo Zheng. 2024. Calibration-compatible Listwise Distillation of Privileged Features for CTR Prediction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24).*

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 452–456.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[15] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten De Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the sixth ACM international conference on Web search and data mining.* 183–192.

[16] Jianqiang Huang, Ke Hu, Qingtao Tang, Mingjian Chen, Yi Qi, Jia Cheng, and Jun Lei. 2021. Deep Position-wise Interaction Network for CTR Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21).*

[17] 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.

[18] Xiang Li, Shuwei Chen, Jian Dong, Jin Zhang, Yongkang Wang, Xingxing Wang, and Dong Wang. 2023. Decision-Making Context Interaction Network for Click-through Rate Prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence.*

[19] Xiao Lin, Xiaokai Chen, Linfeng Song, Jingwei Liu, Biao Li, and Peng Jiang. 2023. Tree based Progressive Regression Model for Watch-Time Prediction in Short-video Recommendation. *arXiv preprint arXiv:2306.03392* (2023).

[20] Yuhan Quan, Jingtao Ding, Chen Gao, Nian Li, Lingling Yi, Depeng Jin, and Yong Li. 2023. Alleviating Video-length Effect for Micro-video Recommendation. *ACM Transactions on Information Systems* 42, 2 (2023), 1–24.

[21] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[22] Shisong Tang, Qing Li, Dingmin Wang, Ci Gao, Wentao Xiao, Dan Zhao, Yong Jiang, Qian Ma, and Aoyang Zhang. 2023. Counterfactual Video Recommendation for Duration Debiasing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 4894–4903.

[23] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17.* 1–7.

[24] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* 115–124.

[25] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the eleventh ACM international conference on web search and data mining.* 610–618.

[26] Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. 2016. Using Spearman's correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience* 28, 14 (2016), 3866–3878.

[27] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged features distillation at taobao recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2590–2598.

[28] Han Xu, Taoxing Pan, Zhiqiang Liu, Xiaoxiao Xu, and Lantao Lu. 2024. Incorporating Group Prior into Variational Inference for Tail-User Behavior Modeling in CTR Prediction. arXiv:2410.15098

[29] Han Xu, Hao Qi, Yaokun Wang, Pei Wang, Guowei Zhang, Congcong Liu, Junsheng Jin, Xiwei Zhao, Zhangang Lin, Jinghe Hu, and Jingping Shao. 2023. PCDF: A Parallel-Computing Distributed Framework for Sponsored Search Advertising Serving. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track: European Conference, ECML PKDD 2023.*

[30] Xiaoxiao Xu, Chen Yang, Qian Yu, Zhiwei Fang, Jiaxing Wang, Chaosheng Fan, Yang He, Changping Peng, Zhangang Lin, and Jingping Shao. 2022. Alleviating Cold-start Problem in CTR Prediction with A Variational Embedding Learning Framework. In *Proceedings of the ACM Web Conference 2022.* 27–35.

[31] Shuo Yang, Sujay Sanghavi, Holakou Rahmanian, Jan Bakus, and Vishwanathan SVN. 2022. Toward Understanding Privileged Features Distillation in Learning-to-Rank. *Advances in Neural Information Processing Systems* 35 (2022), 26658–26670.

[32] Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, Dong Zheng, Peng Jiang, and Kun Gai. 2022. Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 4472–4481.

[33] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 43–51.

[34] 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. 5941–5948.

[35] 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.

Table 7: Evaluation results on Avito-Expo and Video-Expo datasets. Both mean values and standard deviations are presented.

method	Kuai-Expo		Avito	
	AUC	GAUC	AUC	GAUC
DNN	79.36%±0.017%	75.44%±0.014%	77.37%±0.013%	71.58%±0.009%
DNN with VKD	79.37%±0.022%	75.40%±0.017%	77.63%±0.011%	71.86%±0.008%
DNN with MIKD	79.49%±0.017%	75.51%±0.018%	77.84%±0.023%	71.88%±0.011%
DCN	79.11% ±0.026%	75.15%±0.022%	77.45%±0.014%	71.69%±0.012%
DCN with VKD	79.37%±0.025%	75.39%±0.023%	77.55%±0.013%	71.74%±0.010%
DCN with MIKD	79.44%±0.012%	75.42%±0.012%	77.82%±0.016%	71.96%±0.015%
DeepFM	79.29 ±0.029%	75.36%±0.030%	77.59%±0.014%	71.74%±0.011%
DeepFM with VKD	79.41%±0.021%	75.43%±0.025%	77.76%±0.023%	71.95%±0.018%
DeepFM with MIKD	**79.59%±0.020%**	**75.64%±0.017%**	**78.27%±0.019%**	**72.12%±0.014%**