Alleviating Video-Length Effect for Micro-video Recommendation

YUHAN QUAN, Tsinghua University, China
JINGTAO DING*, Tsinghua University, China
CHEN GAO, Tsinghua University, China
NIAN LI, Tsinghua University, China
LINGLING YI, Tencent, China
DEPENG JIN, Tsinghua University, China
YONG LI, Tsinghua University, China

Micro-videos platforms such as TikTok are extremely popular nowadays. One important feature is that users no longer select interested videos from a set, instead they either watch the recommended video or skip to the next one. As a result, the time length of users' watching behavior becomes the most important signal for identifying preferences. However, our empirical data analysis has shown a video-length effect that long videos are easier to receive a higher value of average view time, thus adopting such view-time labels for measuring user preferences can easily induce a biased model that favors the longer videos. In this paper, we propose a **V**ideo **L**ength **D**ebiasing **Rec**ommendation (VLDRec) method to alleviate such an effect for micro-video recommendation. VLDRec designs the data labeling approach and the sample generation module that better capture user preferences in a view-time oriented manner. It further leverages the multi-task learning technique to jointly optimize the above samples with original biased ones. Extensive experiments show that VLDRec can improve the users' view time by 1.81% and 11.32% on two real-world datasets, given a recommendation list of a fixed overall video length, compared with the best baseline method. Moreover, VLDRec is also more effective in matching users' interests in terms of the video content.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Debias, Micro-video Recommendation, Multi-task Learning

1 INTRODUCTION

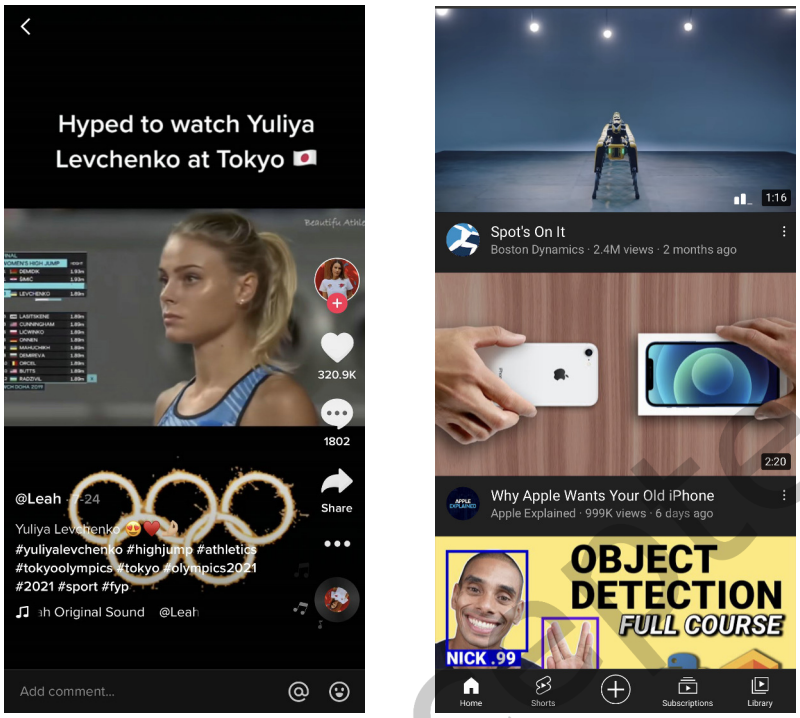
Recommender systems, which can provide items that users may be interested in from a large number of item candidates in a personalized way, are widely deployed nowadays for filtering information or content. Recently, with the help of recommender system, micro-video platforms such as TikTok[1] have swept the world. In fact, there is a significant gap between the micro-video recommender system with the traditional video platforms, such as YouTube recommendation [7, 8][2]. In traditional video websites, a user is always shown/recommended a

*corresponding author
[1]https://www.tiktok.com
[2]In fact, YouTube's iOS App has also been added a tab for micro-video recommendation recently.

Authors' addresses: Yuhan Quan, quanyh15@tsinghua.org.cn, Tsinghua University, Beijing, China; Jingtao Ding, dingjt15@tsinghua.org.cn, Tsinghua University, Beijing, China; Chen Gao, chgao96@gmail.com, Tsinghua University, Beijing, China; Nian Li, linian21@mails.tsinghua.edu.cn, Tsinghua University, Beijing, China; Lingling Yi, chrisyi@tencent.com, Tencent, Shenzhen, China; Depeng Jin, jindp@tsinghua.edu.cn, Tsinghua University, Beijing, China; Yong Li, liyong07@tsinghua.edu.cn, Tsinghua University, Beijing, China.

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.
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1046-8188/2023/8-ART $15.00
https://doi.org/10.1145/3617826

Just Accepted



(a) (b)

Fig. 1. (a)TikTok (micro video app), showing only one video at a time. (b)Youtube (video app), showing multiple videos at a time, *i.e.*, users need to click on the video of interest to watch.

Just Accepted

list of videos, and then he/she can select and click one video he/she feels interested in. As we have mentioned above, the new micro-video platform has completely upgraded the pipeline, of which the most representative one is TikTok. As shown in Fig. 1(a)(b), compared with YouTube, the video length is relatively shorter, generally from tens of seconds to several minutes. The brand new paradigm of user-video interaction can be described as: the videos keep continuously playing until the user slides down the screen, after which another video shows in a similar way.

This difference in interaction manner has actually caused significant changes in the recommender systems. Specifically, for traditional video recommender systems, the main optimization goal is to increase the user's click-through rate (CTR). If the user clicks on a video, it can be considered that the user is interested in this video and can be seen as a positive signal [7, 25]. However, in the brand new micro-video recommender system, users *do not click anymore* since the videos are automatically exposed and played. It means the recommender system can no longer obtain the user's interest through the "click" behavior. Instead, it can only collect a new kind of feedback, the time length that the user watched the video.

An intuitive solution is to first train the model by predicting user-video view time length and then recommending by ranking the predicted results from long ones to short ones. However, users' interactions with longer-length videos are naturally easier to reach longer view time, which results in a longer video will be more likely to be recommended with such an intuitive solution. We call this commonly existing phenomenon in micro-video

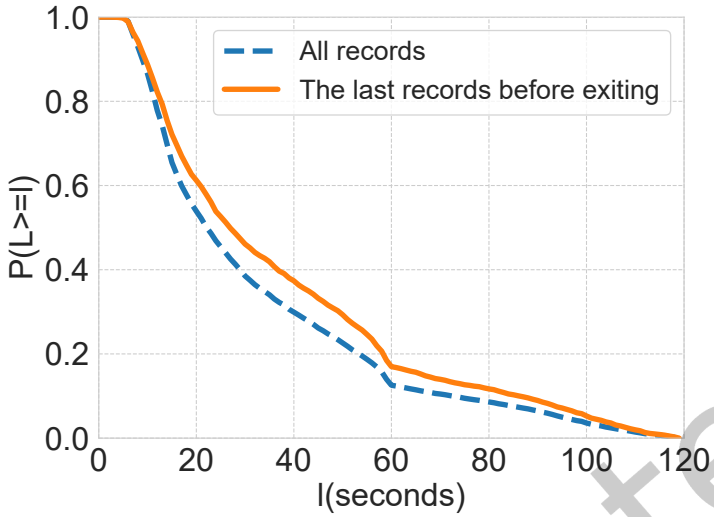


Fig. 2. Distribution of video length for all viewing records and the last viewing records before exiting the app (on wechat dataset).

Just Accepted

platforms as *video-length bias*, where the longer-length videos are particularly favored in the above view-time oriented recommendation scenario. Although it seems acceptable for the platform to recommend longer videos and receive a generally higher value of total view-time, the user engagement might be harmed due to the following two reasons. First, long videos are easier to cause user fatigue, which has been confirmed by our empirical data analysis of users' exit behavior in a micro-video platform, Wechat[3] Channels. As shown in Fig. 2, the last watched video before a user exits the platform tends to be longer than other ones. Second, this unfair advantage of long videos during the learning process possibly includes those undesirable or low-quality videos in the major group, which is known as bias amplification and further hurts recommendation accuracy [39]. Therefore, in this paper, we focus on alleviating the above video-length bias for micro-video recommendation.

Similar to previous studies of popularity bias and position bias in recommender systems [1, 15], the collected data with video-length bias also exhibits the distorted user preferences, where longer view time might be caused by longer video length instead of user satisfaction. However, this new research problem faces two unique challenges that rule out the off-the-shelf debias solutions in other problems like popularity bias and position bias. **First, the continuous characteristics of both video length and view time largely increase the modeling difficulty of unbiased learning[4].** On the one hand, characterizing the effect of a continuous attribute, *i.e.*, video length in our case, generally requires a suitable quantization to avoid data sparsity. On the other hand, learning with the continuous view-time label may suffer from the extreme values that worsen the variance issue of previous unbiased learning methods like inverse propensity scoring (IPS) [2, 19]. **Second, the complex relation between video length and view time makes it non-trivial to identify true user preferences.** Generally, in previous studies, the bias factor either impacts the observation probability of a specific item, as in cases of popularity bias or position bias, or correlates with users' intrinsic interests among different item groups, as in the phenomenon of bias amplification. However, in our case of video-length bias, the bias factor, *i.e.*, the video length, directly

[3]https://www.wechat.com/en
[4]Video length and view time both have a time unit of millisecond, and they are almost continuous.

impacts the measurement of the preference indicator, *i.e.*, the view time, making it challenging to define a proper label that can represent user preferences in view-time oriented recommendation scenarios.

In this paper, we propose a micro-video recommendation approach named **V**ideo **L**ength **D**ebiasing **Rec**ommendation (short for VLDRec) to alleviate the video-length effect. To overcome the first challenge of modeling continuous video-length effect and continuous view-time labels, we first devise a suitable video grouping based on video length, which is motivated by an important data observation that videos with similar time lengths have a similar distribution of completion rate. Then we adopt a simple workaround to continuous label issues by following a learning-to-rank modeling framework. In terms of the second challenge in handling the complex relation between video length and view time, we leverage two bias-alleviating data labeling approaches that can better capture users' real preferences regardless of video length. Combined with a length-conditioned sample generation module and a multi-task user preference learning strategy, our proposed VLDRec can achieve an undistorted model training process with the collected view data under a severe video-length bias. Moreover, besides the model training, to ensure fair comparisons regardless of the observed biased video length effect, VLDRec further incorporates a simple but undistorted Top-T metric for evaluating model performance in the micro-video recommendation.

To summarize, the major contributions of this work are as follows:

- Different from traditional recommender systems, we approach the brand new problem in the micro-video recommendation, of which the biased video length effect widely exists and may worsen the recommendation performance.
- Motivated by empirical observations regarding the relationship between video length and view time, we propose a novel and general micro-video recommendation method including three parts of bias-alleviating data labeling, length-conditioned sample generation and multi-task user preference learning, which further incorporates a length-invariant Top-T evaluation metric to alleviate the video length effect in both modeling training and evaluation.
- We conduct extensive experiments on both public and industrial datasets, and the experimental results demonstrate the effectiveness of our proposed VLDRec method in terms of both longer view time and higher user interest fitness.

The rest of this paper is as follows. We discuss the related works in Section 2. In Section 3, we present the motivation from real-world data and formulate the research problem. We then introduce our proposed VLDRec method in detail in Section 4. We conduct experiments in Section 5. Finally, in Section 6, we conclude this paper and discuss the future works.

2 RELATED WORKS

2.1 Debias in Recommendation

In the recommender system, bias has a variety of sources [5]. Common bias includes popularity bias [27] which caused by popular items being more exposed, selection bias [28] which caused by the fact that users only select items they are interested in, exposure bias [11] which caused by the fact that users can only interact with exposed items, etc. To alleviate these biases, Inverse propensity scoring (IPS) [2, 19, 29, 32] and its improved methods [4, 16] are the most commonly used debiasing methods, which main idea is lowering the weight of the items which have advantages in the recommended results. Therefore, the model will be less influenced by biased instances. In addition, for the bias caused by missing or noisy data, it is usually solved by the relabel method, including heuristic [34] or model-based methods [30, 43]. Also, IPS and relabel methods can be combined [36]. Besides, there are some recently proposed methods, such as disentangling method [26, 42, 54], counterfactual method [44] or causal graph based method [40, 46, 51]. These methods alleviate the bias by designing unbiased training targets or instances, or adjusting the prediction results of the model based on mathematical principles. These methods all have some effect on the biases they deal with. Although there has been a lot of related work on debiasing

Just Accepted

recommender system, most of these bias will only affect the display of recommendation results. However, this paper focuses on a different problem where the video length directly affects the label of the samples. Another angle of analyzing possible impact of video-length effect is fairness [22], as the biased modeling among videos of different video-length can induce unfair exposure favoring long videos [21, 37].

Moreover, we notice that there are some concurrent works that proposes a debiasing recommendation method to handle duration bias in micro-video recommender systems [49, 53]. Although this duration bias problem is similar to the aforementioned video length effect, we propose to alleviate the biased effect by following the idea of regularization [5], *i.e.*, learning intrinsic user preferences from less biased feedback data, while above concurrent work [49] follows the idea of causal inference and another work [53] propose a new unbiased prediction objective remove bias and optimize this objective by adversarial learning.

In general, compared to the related work on debiasing in recommender systems, our work follows a data-driven paradigm, designing bias alleviating method motivated by empirical observations instead of directly relying on theories like causal inference, and is optimized for user viewing time goals, which is a key performance indicator in micro-video recommendations. Besides learning method, we further incorporate a length-invariant Top-T evaluation metric to alleviate the video length effect in both model training and evaluation.

2.2 Video Recommendation

Video recommendation is an important topic in the recommender system. In terms of content and user interface, there are some differences between video recommendation and recommender systems suitable for e-commerce, news or other scenarios. On the one hand, due to the large amounts of multimedia information and features in videos, some works try to effectively utilize the visual or multimedia features in videos [6, 48]. These works focus on how to extract effective features or mix multimodal features. On the other hand, in the scenario of video recommendation, some user behaviors and patterns are different from other scenarios, such as accidentally watching behavior [45], dynamic interest [24, 25], purchase intention through disseminating micro-videos [20], multimodal interests [14, 38] and other implicit and explicit feedback [10, 35].

Although there have been various works related to video recommendation, our work does not focus on how to extract and utilize the multimedia features of videos, or user behaviors. We mainly focus on the biased video length effect in recommendation due to changes in the user interface in micro-video platforms and try to alleviate this bias as much as possible, so the recommendation model can accurately identify user preferences.

3 PROBLEM FORMULATION AND DATA OBSERVATION

We first give the problem formulation of the micro-video recommendation that mainly aims to maximize users' view time. Then we conduct a preliminary analysis on collected user-video interaction data, which motivates our design of the proposed VLDRec method.

3.1 Problem Formulation

Let $U = \{u_1, u_2, ..., u_M\}$ and $V = \{v_1, v_2, ...v_N\}$ denote a set of M users and a set N videos, respectively. The video length of v_j is represented as l_j. For each user $u_i \in U$, given the set of his/her historical interactions S_{u_i}, each $(v_k, t_{ik}) \in S_{u_i}$ representing u_i has watched v_k with a length of t_{ik}, the target is to recommend a new video $v_j \in \{v_k | v_k \in V \wedge v_k \notin S_{u_i}\}$ with the highest view time $\hat{t}_{ij}$. Specifically, we formulate this task as a learn-to-rank problem by learning a scoring function $f(u_i, v_j)$ to indicate u_i's preference on v_j, which measures u_i's willingness to watch v_j with a long time t_{ij} instead of skipping to a next recommended video. When user u_i watches a video v_k with time t_{ik}, the triplet (u_i, v_k, t_{ik}) is defined as a sample. If $p_{ik} \geq 1$, then the sample is a completed sample.

The notations we use and their descriptions are shown in Table 1.

Just Accepted

Table 1. The description of notations.

Notations	Description
U, V	Set of users and videos.
M, N	The size of user sets and video sets.
u_i, v_j	The specific user and video.
$\mathbf{u}_i$, $\mathbf{v}_j$	The embedding of specific user and video.
l_j	The length of video j
t_{ij}	The view time of video j watched by user i
p_{ij}	The play progress of video j watched by user i
S_{u_i}	The set of videos watched by user i
g_v	The v-th group of video sets
v_k^{un}	Video k sampled from the group of positive instance
τ	Threshold for pointwise hard labeling
α, β	Hyper-parameter of multi-task learning and sampling
θ	Model parameter
L_{BPR}	The BPR loss function
f	The preference score prediction function for users and videos
$\hat{\Psi}_{uni}$	The uniform sampler
$S^-_{u_i,v_j}$	The set of videos watched by user i with shorter view time compared with (u_i, v_j)

3.2 Data Observation

In order to investigate and understand user behaviors in micro-video recommendation scenarios, we choose two large-scale datasets collected from two leading micro-video platforms, *i.e.*, Kuaishou[5] and Wechat Channels[6] We leave the detailed descriptions of the above two datasets in Section 5. Here we focus on two indicators of user engagement on recommended micro-videos, *i.e.*, view time and completion rate [52]. The former is intuitive, but we will highlight its potential bias in favoring longer videos by illustrating its relationship with video length in Fig. 3(a) and (b). Comparatively, the latter counts the proportion of completed plays for a specific video, which is normalized into $[0, 1]$ by considering both view time and video length. Mathematically, the completion rate of v_j is equal to the number of completed samples (*i.e.*, $t_{\bullet j} \geq l_j$) divided by the total number of samples corresponding to the video, which is as follows,

$$Completion_rate_v = \frac{\#completed_samples_v}{\#All_samples_v}. \tag{1}$$

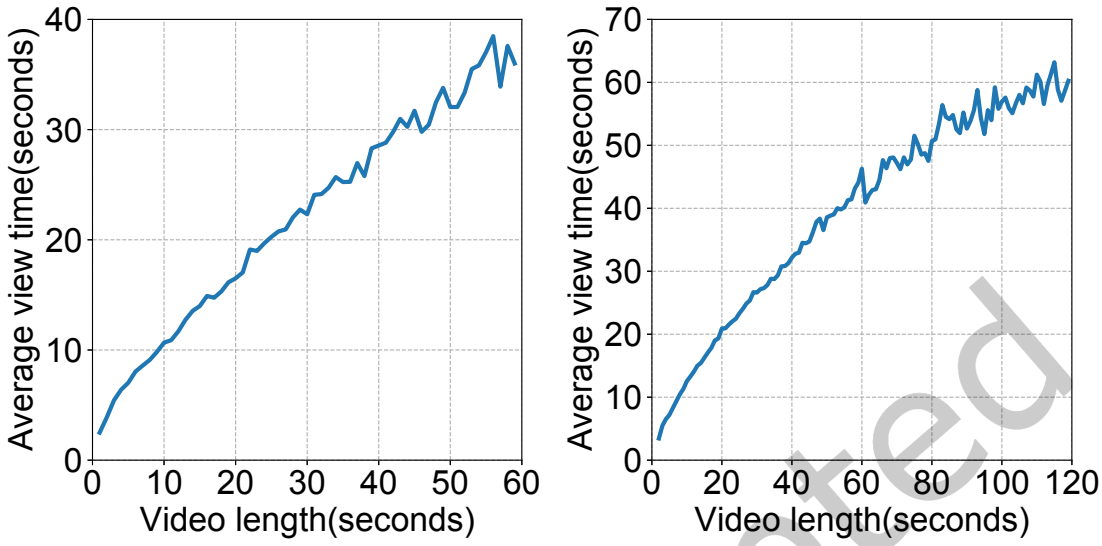
In Fig. 4, by analyzing the distribution of completion rate ($p25$, $p50$ values) for micro-videos with the same video length, we demonstrate that this metric is much fair when comparing users' preference among micro-videos with similar video length.

Specifically, our preliminary analysis has the following two key observations.

- **Long videos are much easier to receive a higher value of average view time.** As shown in Fig. 3(a) and (b), real-world data in both KuaiShou and Wechat follow a similar pattern that the average view time is nearly linear to the increasing video length. Consequently, view time is a biased metric when measuring a user's

[5]https://www.kuaishou.com/en
[6]https://channels.weixin.qq.com.

Just Accepted



(a) Kuaishou (b) Wechat

Fig. 3. Average view time of videos of different length.

Just Accepted

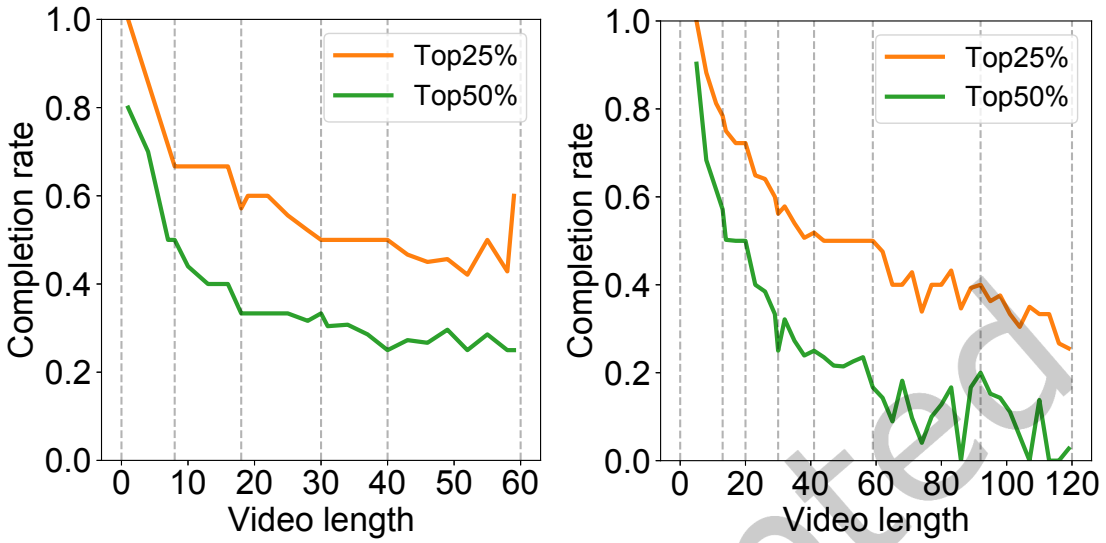
preference on a micro-video without considering the specific video length. This can lead to severe performance degradation when we attempt to capture user preference with traditional approaches like regression.

- **Videos with similar time lengths have a similar distribution of completion rate.** As shown in Fig. 4(a) (KuaiShou), by illustrating the $p75$ (the third quartile, *i.e.*, ranked at the top 25%) and $p50$ (the median) distribution values of micro-videos ranging from 1s to 59s, we can roughly divide them into five groups according to video length, with the flat curve within each group representing a similar distribution pattern. Specifically, these five groups are $[1-8s, 8-18s, 19s-30s, 31s-40s, 41-59s]$. For example, we can observe that micro-videos ranging from 30s to 40s tend to have a completion rate of about 0.5 ($p75$). Similarly for Wechat Channel dataset (Fig. 4(b)), the suitable grouping is $[0-13s, 14-20s, 21-30s, 31-41s, 42-59s, 60-92s, 93-120s]$. In a word, to capture unbiased user preference from their viewed micro-videos, one needs to extract pairwise ranking relations conditioned on the video length.

In short, there exists a bias of video-length in video-watching behaviors.

4 METHOD

The proposed VLDRec method models user preferences in a view-time oriented learning-to-rank manner, where two specific labeling approaches are designed to alleviate the video-length effect that can distort the learned user preference. Motivated by previous analysis regarding to the biased preference signal of view time, VLDRec further integrates a length-conditioned sample generation module, which is jointly optimized via multi-task learning. Moreover, to better evaluate micro-video recommendation models, VLDRec adopts a simple but effective Top-T evaluation metric that can make fair comparisons regardless of the observed biased video length effect. The overall framework of VLDRec is shown in fig 5.



(a) Kuaishou (b) Wechat

Fig. 4. The relationship between completion rate and video length (aggregated statistics every one second).

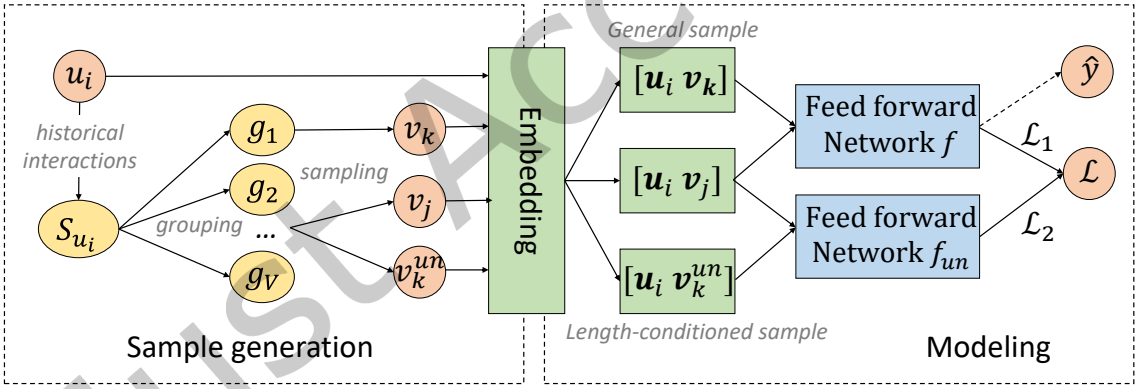


Fig. 5. Overall framework of VLDRec

Just Accepted

4.1 Labeling Approach for View-time Oriented Learning-to-rank

For the micro-video recommendation problem defined in Section 3.1, we adopt a learning-to-rank approach that trains a model to recommend micro-videos with a large probability of generating user engagements, *i.e.*, views.

Generally, to learn recommender models from implicit feedback, Rendle *et al.* [31] proposed the Bayesian Personalized Ranking (BPR) method, which assumes that a positive instance should be predicted with a much higher score over the negative one. Based on BPR, the training objective of the recommender model can be formulated as minimizing the following loss function:

$$L_{BPR} = \sum_{(u_i,v_j)\in S_{u_i}} -\ln \sigma(f(u_i,v_j) - f(u_i,v_k)),$$
$$\text{where } v_k \sim \hat{\Psi}_{uni}(S^-_{u_i,v_j}). \tag{2}$$

For each user u_i, the predicted preference score on micro-videos is denoted as $f(u_i, \bullet)$. The negative instance v_k is generated by a uniform sampler $\hat{\Psi}_{uni}$ that takes the candidate set $S^-_{u_i,v_j}$ as an input, while the positive instance i is randomly chosen from ground truth set S_{u_i}. Minimizing L_{BPR} is equivalent to maximizing the margin between $f(u_i, v_j)$ and $f(u_i, v_k)$, which encourages the recommender to learn the pairwise ranking relation of user preference between v_j and v_k.

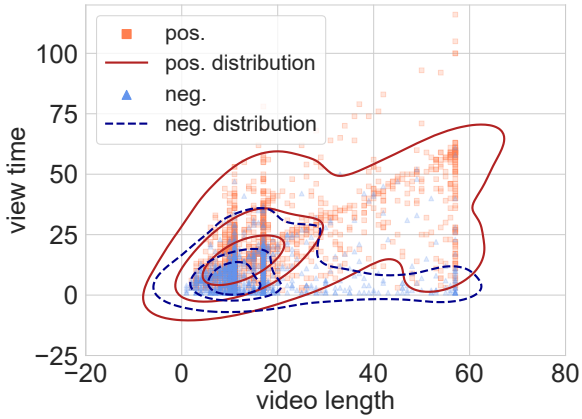
Therefore, an intuitive solution for solving the micro-video recommendation problem is directly putting the instances with longer view time ahead of those with a shorter length. Specifically, when $S^-_{u_i,v_j}$ is the set of past instances with shorter view time than current positive instance (u_i, v_j), *i.e.*, $\{(u_i, v_k)|t_{ik} < t_{ij}\}$, the trained model is able to recommend micro-videos that may be watched longer by users. However, as we discussed in the preliminary analysis, the indicator of view time suffers from the biased video length effect by favoring longer videos, which may cause the distortion of learned user preference.

In our proposed VLDRec model, we instead choose to measure user preference based on another indicator of play progress, *i.e.*, $p_{ij} = t_{ij}/l_j$. Specifically, we adopt two different definitions of progress-based labels. The first is pointwise hard labeling depending on whether p_{ij} exceeds a certain threshold τ. Motivated by the previous observation that videos with similar time lengths have a similar distribution of completion rate, we set τ as $\tau(l_j)$, relevant to the video length l_j. According to our analysis in Fig. 4, the micro-videos can be divided into several groups based on their length, *i.e.*, $\{g_1, g_2, ...g_V\}$, where V is 5 (KuaiShou) and 7 (WeChat Channel), respectively. For these two datasets, we decided the boundary value of video length in each group by checking whether at least one of Top-25% and Top-50% curves in Fig. 4 would go through a significant change (*i.e.*, from decreasing to non-decreasing or from increasing to non-increasing) at this point compared with neighboring points This operation is performed manually. Thus we allow videos within the same group sharing the same $\tau = \tau(l_j) = \tau(g)$. Without loss of generality, $\tau(g)$ for each group is set as $p80$ value of the play progress distribution. In other words, the top 20% samples ranked by their play progress values are considered as positive instances under this pointwise hard labeling.

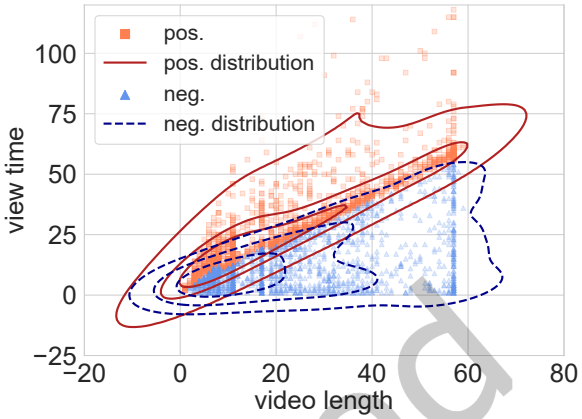
Besides, we also use another pairwise margin-based labeling. It sets a constraint that there should exist a margin ϵ between the progress values of a positive instance (u_i, v_j) and a negative instance (u_i, v_k), *i.e.*, satisfying $p_{ij} - p_{ik} > \epsilon$. Compared with the above pointwise hard labeling, this pairwise labeling focuses on comparing two samples belonging to the same user, which is more friendly to users with rich interaction history. Therefore, we leverage the advantages of both two approaches by switching between them, which is controlled by hyper-parameters β.

In Fig. 6, we plot the positive and negative sample distribution under the above three labeling strategies, respectively, in terms of both video length (x-axis) and view time (y-axis). Specifically, we uniformly sample 2,000 training samples in one epoch to obtain the population. As illustrated in Fig. 6(a), negative samples tend to be located in the bottom-left corner, *i.e.*, with both short video length and short view time. While for our proposed two strategies, the distribution of negative samples extends from bottom left to upper right, indicating a better discriminative capability of the learned model, as videos with long duration and long view time can also be possibly chosen as negative samples. Compared with the pointwise hard labeling (Fig. 6(b)), the pairwise margin-based labeling (Fig. 6(c)) generates more similar distributions among positive samples and negative samples, which both have multiple centers of both positive and negative samples, and can serve as a complement to the former.

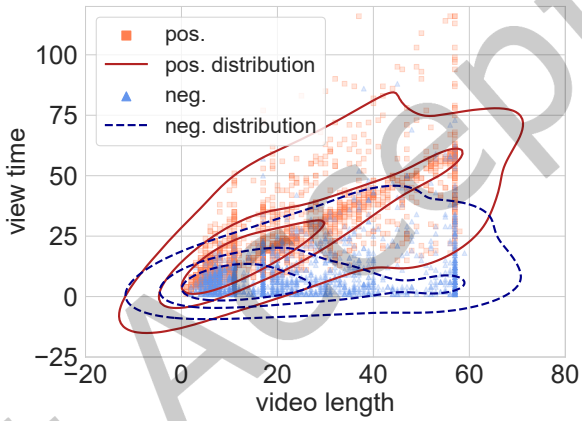
Just Accepted



(a) Only based on view time



(b) Based on pointwise hard labeling



(c) Based on pairwise margin-based labeling

Fig. 6. Correlations between video length and view time of positive (or negative) samples under different labeling strategies.

Just Accepted

4.2 Length-conditioned Sample Generation

Our previous observation shows that completion rate distribution stays stable conditioned on the video length. Thus it is reasonable to draw a conclusion that, among two micro-videos with similar time length, a user favors the one with higher play progress (*i.e.*, longer view time). Inspired by this, we are able to alleviate biased video length effect by learning pairwise rank relations among videos within the same group, which is achieved by a length-conditioned sample generation module. Specifically, for each sample (u_i, v_j), we additionally choose a video v_k^{un} from the group corresponding to video v_j to construct another training pair. The labels of v_j and v_k^{un} are defined similarly as introduced before.

The complete process of generating training samples is shown in Algorithm 1, which contains two parts of samples, one with video length effect and the other with alleviated effect. Specifically, if current training pair uses the pointwise hard labeling, the candidate sets for sampling $S^-_{u_i,v_j}$ are $\{(u_i, v_k)|(t_{ik} < \tau(g) \wedge t_{ij} > \tau(g)) \vee (t_{ik} >$

Algorithm 1 Length-conditioned Sample Generation Algorithm

Require: Positive instance (u_i, v_j), where $v_j \in g_v$ and hyper parameter β, τ, ϵ

```
p = rand()
if p < β then
    //General sample
    sample v_k from S_{u_i} where (t_ik < τ(g) ∧ t_ij > τ(g)) ∨ (t_ik > τ(g) ∧ t_ij < τ(g))
    // Length-conditioned
    sample v_k^un from S_{u_i} ∩ g_v where (t_ik < τ(g_v) ∧ t_ij > τ(g_v)) ∨ (t_ik > τ(g_v) ∧ t_ij < τ(g_v))
else
    //General sample
    sample v_k from S_{u_i} where |p_ij − p_ik| > ε
    // Length-conditioned
    sample v_k^un from S_{u_i} ∩ g_v where |p_ij − p_ik| > ε
end if
return Positive instance (u_i, v_j) and negative instances (u_i, v_k), (u_i, v_k^un)
```

$\tau(g) \wedge t_{ij} < \tau(g))\}$ and $\{(u_i, v_k^{un}) | (t_{ik} < \tau(g) \wedge t_{ij} > \tau(g) \wedge v_k^{un} \in g(v_j)) \vee (t_{ik} > \tau(g) \wedge t_{ij} < \tau(g) \wedge v_k^{un} \in g(v_j))\}$. Otherwise, current training pair uses the pairwise margin-based labeling, and $S^-_{u_i,v_j}$ are $\{(u_i, v_k) || p_{ik} - p_{ij}| > \epsilon\}$ and $\{(u_i, v_k^{un}) || p_{ik} - p_{ij}| > \epsilon \wedge v_k^{un} \in g(v_j)\}$.

4.3 Multi-task User Preference Learning

In this part, we design a multi-task learning model that enables joint optimization with two parts of samples. Specifically, it contains a shared embedding module and two independent feedforward neural networks (FNN) used for learning and predicting user preference from two types of samples, respectively. First, for instances (u_i, v_j), (u_i, v_k),(u_i, v_k^{un}), the shared embedding module outputs $(\mathbf{u}_i, \mathbf{v}_j)$, $(\mathbf{u}_i, \mathbf{v}_k)$ and $(\mathbf{u}_i, \mathbf{v}_k^{un})$ through a embedding map. These embeddings are further constructed into two training pairs, *i.e.*, $\{(\mathbf{u}_i, \mathbf{v}_j), (\mathbf{u}_i, \mathbf{v}_k)\}$ and $\{(\mathbf{u}_i, \mathbf{v}_j), (\mathbf{u}_i, \mathbf{v}_k^{un})\}$. Then, the corresponding FNN for each training pair can generate the preference scores that are used for training or prediction. Note that these two networks do not share model parameters, and can be any commonly used recommendation models like NFM [18], DeepFM [17], AutoInt [33] etc. Finally, a multi-task learning based training process is conducted to jointly learn user preference from both two parts of training samples, *i.e.*, one with biased observation affected by video length and the other with debiased manipulation. Here we use the aforementioned BPR loss [31], and then add the two losses by linear weighting to get the final loss and use it for backpropagation. The formula is as follows,

$$\begin{aligned} L_1 &= L_{BPR}(f(\mathbf{u}_i, \mathbf{v}_j), f(\mathbf{u}_i, \mathbf{v}_k)), \\ L_2 &= L_{BPR}(f_{un}(\mathbf{u}_i, \mathbf{v}_j), f_{un}(\mathbf{u}_i, \mathbf{v}_k^{un})), \\ L &= \alpha * L_1 + (1 - \alpha) * L_2, \end{aligned} \tag{3}$$

where f and f_{un} respectively denote the model used for two parts of training data.

So far, we have completed the training process of VLDRec, and the overall procedure is shown in Algorithm 2.

4.4 Length-invariant Top-T Evaluation Metric

In both literatures and practices on personalized recommender systems [41, 47], Top-K based metrics are widely used for evaluating models, such as *Recall@K* or *NDCG@K*. However, in micro-video recommendation scenarios, as mentioned above, users can only watch one video at a time and the video is automatically played, so the

Just Accepted

Algorithm 2 Overall procedure of VLDRec

Require: U, V, $\{l_j\}$ and training instances $\{u_i, v_j, t_{ij}\}$, Randomly initialize θ
1: calculate $Completion_rate_v$ for $v \in V$
2: generate video groups $\{g_v\}$
3: **while** *Stopping criteria is not met* **do**
4: generate negative instances (u_i, v_k), (u_i, v_k^{un}) by Algorithm 1 for positive instance (u_i, v_j)
5: calculate L_1, L_2 and L by Equation 3
6: update model parameter θ by minimizing L
7: **end while**

objective is to maximize users' total view time, which depends on whether the recommendation model can capture the real user preference instead of the distorted one based on biased observations (*i.e.*, longer videos receive longer view time.). Specifically, imagine an extreme case where a model always recommends micro-videos with a long length, a user's $View_Time@K$, which is defined as

$$View_Time@K = \sum_{j=1}^{K} t_{ij}, \tag{4}$$

may still be fairly good even if she completes none of the recommended micro-videos. Therefore, we argue that Top-K based metric is not suitable for view-time oriented recommendation tasks.

To better evaluate micro-video recommendation models from a fair angle, we propose a new Top-T recommendation protocol where each time a list of videos are recommended and the total length of these videos is fixed as T, while the list length K can be a variable. Mathematically, the metric $View_Time@T$ is defined as follows,

$$\begin{aligned} View_Time@T &= \sum_{j=1}^{n} t_{ij}, \\ where \sum_{j=1}^{n} l_j &= T. \end{aligned} \tag{5}$$

If the total video length exceeds T, the last video length and the corresponding user's view time will be adjusted proportionally, so that the total video length exactly equals T. Compared with $View_Time@K$, $View_Time@T$ emphasizes the importance of play progress, which penalizes the extreme case where only long videos are recommended but the corresponding play progress is low. However, it is noteworthy that, though similar, $View_Time@T$ is intrinsically different from the average progress metric, as it gives a comprehensive account of both play progress and total view time.

4.5 Discussion

Regarding the video length bias problem in micro video recommendations, as mentioned above, there have been some related works. While the methods used may vary, the common objective is to mitigate the influence of video length on the recommendation model. From the perspective of causal inference, video length is a confounder between user and video. Therefore, the key to alleviating the bias is to alleviate the impact of the confounder. Existing related works attempt to achieve this by employing backdoor adjustment [49] or designing prediction targets independent of video length [53].

In our research, we conducted empirical data analysis and discovered that longer videos tend to have a higher average view time. Interestingly, this finding has also been directly utilized as prior information in two other

Just Accepted

Table 2. Basic information of datasets.

Dataset	Kuaishou	Wechat
#Samples	1,945,502	3,264,803
#Users	9,829	54,595
#Videos	136,317	62,569
Average video length	17.54s	32.97s
Max video length	60s	120s
Average view time	14.78s	26.32s

related works [49, 53]. Consequently, our approach to mitigating the confounding effect involves generating length-conditioned samples to construct unbiased data sets. Notably, this idea aligns with the theoretical support for causal inference, which is also shared by the related works.

5 EXPERIMENTS

We aim to answer the following three research questions (RQ) in experiments.

- **RQ1**: How does our proposed VLDRec model perform compared with state-of-the-art micro-video recommendation methods. More specifically, does VLDRec successfully alleviate the video length effect that troubles common practice in previous solutions?
- **RQ2**: Whether the modules of our model can work well, including the debias sampling strategy and the multi-task learning based model design.
- **RQ3**: Can the VLDRec capture user preference on micro-videos, in terms of other dimensions besides view time?

5.1 Experimental Settings

5.1.1 Dataset and data preprocessing. We conduct extensive experiments on two real-world datasets collected from popular micro-video applications, Kuaishou and Wechat Channels. The Kuaishou dataset is a public dataset[7] that has been used in the previous work [23]. Since the optimization goal of this work is the user's click behavior rather than the viewing time, which is different from our task, we do not use this work as our baseline. Wechat dataset is collected from real industrial scenarios. Besides, we only keep the instances that users have clicked and watched, as those non-click videos are not watched and do not match the optimization objective in the micro-video recommendation task. It should be noted that in the original data, the WeChat dataset contains timestamps but the Kuaishou dataset does not contain them. Therefore, data analysis like Table 2 can only be performed on the WeChat dataset.

It is noteworthy that in micro-video applications, each video is automatically played. Unless the user slides down to the next video, the current video will be played repeatedly. Therefore, samples that are played too many times may be abnormal and we deleted samples where the video is played more than 3 times($p_{ij} > 3$) repeatedly in two datasets. At the same time, according to the difference among the datasets, we deleted a small number of samples where the video length is very long. In the two datasets, the threshold is set to 60 seconds and 120 seconds, respectively.

Table 2 summarizes some basic information about the datasets we used. Since the two datasets do not contain the timestamp, we randomly split 10% as the validation set and 20% as the test set. According to the length of the video, we divided the Kuaishou and Wechat datasets into 5 and 7 groups respectively. More details about

[7]https://github.com/liyongqi67/ALPINE

Just Accepted

grouping have been described in section 4.1. The features we use include *user_id*, *video_id* and *video_length*, as we aim to evaluate the proposed VLDRec model in a general experimental setting.

5.1.2 Baseline. We compare the proposed VLDRec with three categories of baseline methods.

The first category is two regression-based methods that are widely used in industrial micro-video recommender systems.

- **TimeRegression (TReg)**. This is an intuitive method that takes the length of time the user watched the video as the target(t_{ij}), and sorts according to the predicted view time, so as to obtain the final recommendation result.
- **RateRegression (RReg)**. This baseline regresses the view progress of each sample(p_{ij}), and then the predicted view time is obtained by multiplying the predicted score by the video length.

The next category is two ranking-based methods that follow a similar idea of learning-to-rank as ours in solving micro-video recommendation task.

- **TimeRanking (TRank)**. This method learns to rank according to the view time of the instances, where the negative instances are randomly sampled from the videos watched by the user of the positive instance. Based on our proposed multi-task learning framework, only biased datasets are used for ranking training in this method, that is, the weight of multi-task is set to 0. It is a de-generated model that there is no unbiased dataset to help learn unbiased representations, directly ranking the samples on the biased dataset may cause a large bias in the recommended model obtained by training.
- **RateRanking (RRank)**. This method learns to rank according to the play progress of the instances and its sampling strategy is consistent with the *TimeRanking* method. It should be noted that since the prediction score obtained by the ranking method is not an accurate value, it does not need to be multiplied by the video length, but is directly used for ranking and recommendation.

The final category is five unbiased recommendation methods that try to learn an unbiased recommender model by removing the video length effect.

- **IPS** [19, 32]. The inverse propensity score method estimates the bias by re-weight each instance. For those items that have a greater advantage in recommendation due to bias, the IPS method reduces the weight of these instances to achieve a balance in recommendation results. In our experiment, the weight of each instance is set to the inverse of the video length of each instance; that is, the longer the video length, the lower the weight of the instance.
- **IPS-C** [4]. This method uses max capping to limit the value of IPS, so that the range of IPS values is limited, thereby reducing the variance of the score and enhancing the stability of the model
- **IPS-CN** [16]. On the basis of max capping, this method normalizes the value of IPS, so that the variance of the IPS score is further reduced.
- **IPS-CNSR** [16]. On the basis of normalization, this method adds smoothing operations for the value of IPS.
- **CausE** [3]. It is trained through a large biased dataset and a small unbiased dataset. By using two models to model two datasets respectively, and using L2 regularization to constrain the embedding of the two models, so the impact of the bias of the model can be reduced. In our experiment, the unbiased dataset is obtained by sampling from the same group of the video of the positive instance, which is consistent with part of our VLDRec model.
- **DecRS** [39]. The algorithm avoids the impact of confounder by inserting a backdoor adjustment operator into the existing model and solves the problem of infinite sample space through an approximation algorithm.
- **DVR** [53]. This algorithm designs a new unbiased metric named WTG as the prediction target, and trains the model through adversarial learning.

Just Accepted

Table 3. Search range of hyper-paramaters.

Hyper-parameter	Range
lr	$\{0.005, 0.001, 0.0005, 0.0001\}$
$dropout$	$[0, 1]$
α	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$
β	$\{0.1, 0.3, 0.5, 0.7, 0.9\}$

5.1.3 Evaluation Method. Since the proposed VLDRec is mainly aimed at the ranking stage in recommendation, we only use the collected records to construct recommendation lists. For the samples in the test set, we first aggregated them according to *user_id*. Then, for each user, a recommendation list is generated by sorting candidate micro-videos according to prediction scores and selecting the top ones. The final evaluation metrics are obtained by first calculating per-user value and then averaging among all users.

First of all, we mainly use view time metrics in experiments, including $View_Time@T$ and $View_Time@K$, where the former is less affected by biased observations and formally defined in Section 4.4. Although $View_Time@K$ favors longer videos and cannot reflect users' real preferences, we still use this metric to highlight the spurious goodness-of-fit of prevailing solutions.

Moreover, besides the view time, we also leverage category information of micro-videos to measure whether our recommendation reflects users' actual preferences. Specifically, for Top-K recommended videos of each user, we calculate *size of intersection* and Jensen–Shannon divergence (JSD) [12] by comparing them with the actual Top-K viewed videos ordered by view time. The first metric *size of intersection* is in $[0, K]$. As for JSD, it is defined as

$$\mathrm{JSD}(P||Q) = H(\frac{P+Q}{2}) - \frac{1}{2}(H(P) + H(Q))$$

where H is the Shannon entropy, P and Q are distributions. Lower JSD denotes a closer distribution between the recommendation and real data, indicating a better reflection of user preference.

5.1.4 Implementation Detail. For all methods, we use NFM [18], DeepFM [17] and AutoInt [33] as base model. The embedding size for each feature of all methods is set to 8 and the batch size is set to 1024. The number of hidden layers of the deep part of NFM model is set to $\{32, 16\}$. For the regression method, we use MSE loss. In all experiments, we use the Adam optimizer. Hyper parameters such as learning rate and dropout are obtained through grid search. The search ranges are shown in Table 3. The code and dataset are avaliable[8].

5.2 Performance Comparison (RQ1)

5.2.1 Overall Performance. The overall experimental results are shown in the Table 4, Table 5, Table 6 and Table 7 *w.r.t.* $View_Time@T$. From the above results, we have the following observations:

- **VLDRec significantly improves the recommendation performance by alleviating the video length effect and capturing real user preferences regardless of the video length.** Compared with the best-performed baseline, it outperforms by 1.81% and 11.32% *w.r.t.* $View_Time@120$ in Kuaishou and Wechat with NFM as the base model, respectively. At the same time, when using DeepFM and AutoInt as the base model, VLDRec also achieved at least 5.49% and 7.31% improvements *w.r.t.* $View_Time@120$, respectively. It demonstrates that actively removing the biased effects introduced by video length is vital for learning user preferences among different micro-videos.

[8]https://github.com/SpongeBobSquarePants111/VLDRec

Just Accepted

Table 4. Overall performance comparison between our proposed method and baselines on Kuaishou with NFM as base model(RelaImpr is shorthand for Relative Improvement).

Metric	View_time@120	RelaImpr	View_time@240	RelaImpr
TReg	35.78	25.66%	66.69	23.51%
RReg	25.13	78.91%	37.32	120.71%
TRank	40.87	10.01%	78.79	4.54%
RRank	43.94	2.32%	<u>81.93</u>	0.54%
IPS	40.31	11.54%	77.51	6.27%
IPS-C	43.36	3.69%	80.99	1.70%
IPS-CN	41.62	8.02%	79.03	4.23%
IPS-CNSR	41.80	7.56%	79.75	3.29%
CausE	42.14	6.69%	80.11	2.82%
DecRS	<u>44.16</u>	1.81%	71.69	14.90%
DVR	40.22	10.54%	79.47	3.52%
VLDRec	**44.96**	-	**82.37**	-

Table 5. Overall performance comparison between our proposed method and baselines on Kuaishou with DeepFM as base model(RelaImpr is shorthand for Relative Improvement).

Metric	View_time@120	RelaImpr	View_time@240	RelaImpr
TReg	25.01	92.04%	52.13	67.10%
RReg	25.75	86.52%	41.37	110.56%
TRank	<u>45.53</u>	5.49%	<u>83.39</u>	4.46%
RRank	38.27	25.50%	79.31	9.83%
IPS	35.52	35.22%	75.70	15.07%
IPS-C	36.01	33.28%	76.50	13.87%
IPS-CN	35.58	34.99%	76.50	13.87%
IPS-CNSR	37.61	27.71%	79.06	10.18%
CausE	36.19	32.72%	76.84	13.37%
DecRS	37.26	28.90%	63.05	38.16%
DVR	38.01	20.86%	83.38	4.28%
VLDRec	**48.03**	-	**87.11**	-

- **VLDRec better resists the noisy preference signal in biased observation.** Corresponding to our aforementioned problem of using view time based signal, we observe the significant performance degradation with two regression methods that are common practice in most companies, *i.e.*, TimeRegression and RateRegression. Specifically, VLDRec outperforms the best of them by 25.66% in Kuaishou and 137.30% in Wechat *w.r.t.* $View_Time$@120 with NFM as the base model. At the same time, VLDRec outperforms the best of them 92.04% and 17.56% with DeepFM and AutoInt as the base model. The observed huge performance gap demonstrates that directly using observed view time (or play progress) as the label can be biased and lead to distorted preference learning. Contrastively, VLDRec successfully bypasses this by learning pairwise ranking relations of user preferences among micro-videos with similar time lengths.

Just Accepted

Table 6. Overall performance comparison between our proposed method and baselines on Kuaishou with AutoInt as base model(RelaImpr is shorthand for Relative Improvement).

Metric	View_time@120	RelaImpr	View_time@240	RelaImpr
TReg	40.09	17.56%	65.68	21.92%
RReg	22.36	110.78%	35.00	128.80%
TRank	37.37	26.12%	66.16	21.04%
RRank	34.23	37.69%	61.72	29.75%
IPS	41.86	12.59%	71.75	11.61%
IPS-C	40.39	16.69%	69.38	15.42%
IPS-CN	40.38	16.72%	69.53	15.17%
IPS-CNSR	<u>43.92</u>	7.31%	74.90	6.92%
CausE	42.55	10.76%	72.20	10.91%
DecRS	43.18	9.15%	<u>76.02</u>	5.34%
DVR	38.15	19.05%	74.21	7.33%
VLDRec	**47.13**	-	**80.08**	-

Table 7. Overall performance comparison between our proposed method and baselines on Wechat with NFM as base model(RelaImpr is shorthand for Relative Improvement).

Metric	View_time@120	RelaImpr	View_time@240	RelaImpr
TReg	12.52	137.30%	27.16	97.05%
RReg	12.17	144.12%	21.49	149.05%
TRank	23.50	26.43%	48.10	11.27%
RRank	22.84	30.08%	47.01	13.85%
IPS	22.73	30.71%	46.26	15.69%
IPS-C	22.45	32.34%	47.65	12.32%
IPS-CN	23.53	26.26%	47.40	12.91%
IPS-CNSR	23.55	26.16%	47.07	13.70%
CausE	<u>26.69</u>	11.32%	<u>51.59</u>	3.74%
DecRS	24.54	21.06%	43.23	23.80%
VLDRec	**29.71**	-	**53.52**	-

- **VLDRec further improves the informativeness of training samples by selecting training pairs conditioned on video length.** By choosing pairs of micro-videos with similar time lengths, VLDRec not only alleviates the existing bias in observed data but also leverages the advantage of hard negative sampling that is proven to be useful in improving recommendation performance [9, 50]. Intuitively, for a specific sample (u_i, v_j), choosing a negative sample from the same group as v_j is much more informative for model learning than a different group where video length is shorter. Therefore, compared with those unbiased learning baselines including IPS-based, CausE, DecRS and DVR, VLDRec outperforms by a large margin. Specifically, IPS-based methods suffer from the large variance issue, resulting in unstable performance as shown in Table 4, Table 5, Table 6 and Table 7. As for CausE, DecRS, and DVR they perform fairly competitively but cannot beat VLDRec, as it is not elaborately designed for a view time oriented recommendation task.

Just Accepted

5.2.2 Biased Learning of Common Practice. The above performance comparison has demonstrated the superiority of VLDRec in recommending micro-videos that indeed match user interest and thus generate user engagement. In this part, we further answer the question on how VLDRec alleviates the video length effect that existed in previous solutions.

We first present the recommendation performances of each video-length group (*i.e.*, $\{g_v\}$) *w.r.t.* $View_Time@K$. Specifically, for each user, only top-K micro videos belonging to a certain group are recommended and $View_Time@K$ is used for evaluation. As the recommended videos are of a similar time length, $View_Time@K$ is freed from the aforementioned problem of favoring longer videos. As shown in Table 8 and Table 9, surprisingly, we observe that TimeRegression, RateRegression and DecRS perform rather competitively in each video group. Specifically, for Kuaishou Dataset, TimeRegression and RateRegression methods beat VLDRec in all five groups and DecRS beats VLDRec in four groups. As for Wechat Dataset, DecRS beats VLDRec in all groups, while TimeRegression outperforms in longer video groups (60 - 120s) and RateRegression outperforms in shorter ones (0 - 59s). The above observation implies that these methods are able to learn user preferences among micro-videos with similar time lengths, even better than the proposed VLDRec. However, this is in conflict with the overall performance comparison in Table 4, Table 5, Table 6 and Table 7, where regression based methods are reported to suffer from significant performance degradation and DecRS doesn't perform well.

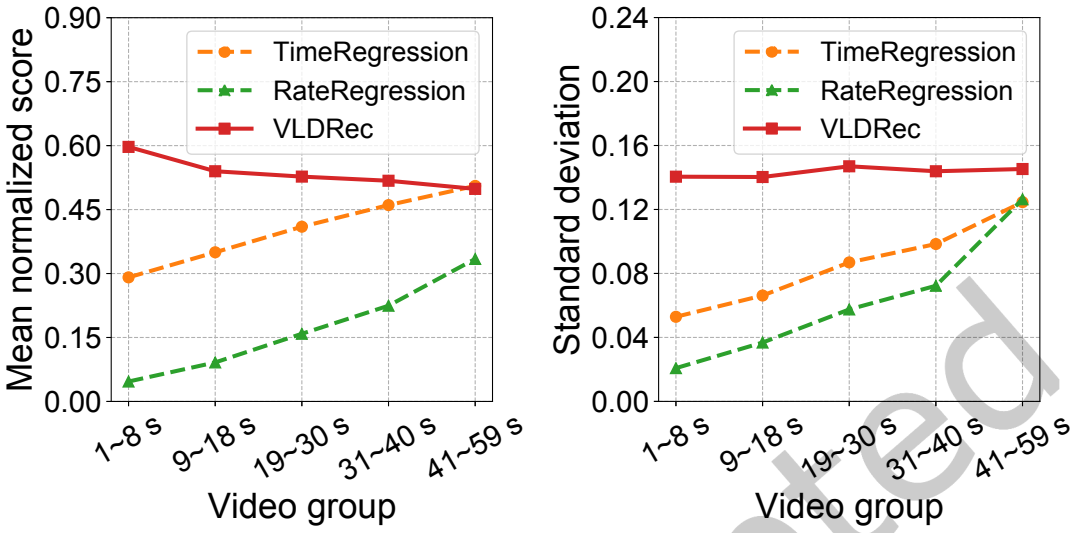
To further explain the above contradictory phenomena, we analyze the distribution of model prediction scores generated by regression methods and VLDRec in Fig. 7 and Fig. 8. Specifically, we normalize the prediction scores of all test set samples generated by each method and then calculate the mean score and standard deviation of each video group. As shown in Fig. 7, we can observe that the average prediction score of TimeRegression and RateRegression increases with the video time length, which is similar to the observation from empirical data that longer videos generally have longer watch time (See Fig. 3.). This indicates a good fit to the training data, which, however, does not guarantee a precise recommendation that matches user preference. Further in Fig. 8, when looking at the variation of scores (*i.e.*, standard deviation), the regression methods still have an increasing standard deviation as the video length increases, and low value of variation in short video groups indicates a centralized prediction distribution. In summary, regression models achieve a good fit of distribution in terms of mean score and their predictions exhibit a rather centralized characteristic in terms of standard deviation, which means they are fairly precise in each video group but cannot overcome the intrinsic problem that longer videos tend to be ranked higher regardless of the actual user preference.

Contrastively, the proposed VLDRec has a relatively stable distribution of both mean score and standard deviation among all video groups. Specifically, VLDRec generates almost equal mean scores among all video groups. Also, its standard deviation is higher than regression models, indicating that a micro-video assumed to be preferred by a certain user can be ranked higher even if this video just lasts less than 10s. Therefore, VLDRec is less influenced by the video length and is able to distinguish micro-videos that users are truly interested in and not interested in. Comparatively, industrial recommender systems normally rely on ranking with a fusion of multiple objectives (like a linear combination of two scores from TimeRegression and RateRegression) to alleviate the video-length effect in their recommendation results [52], which is less elegant and requires a huge manual effort for tuning fusion-related hyperparameters.

5.3 Ablation Study (RQ2)

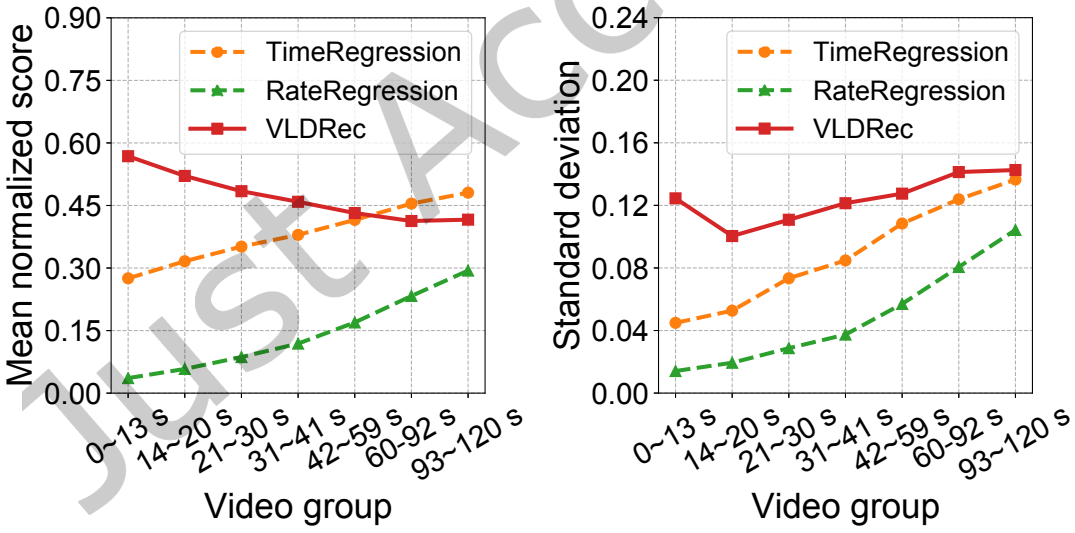
5.3.1 Performance on Different Sampling Strategy. In the process of instance generation, we proposed two sampling methods. The pointwise hard labeling based sampling and the pairwise margin-based labeling based sampling. The former focuses on characterizing the overall distribution between positive instances and negative instances, while the latter strengthens the modeling of preference rankings of the same user. We use hyper-parameter β to control the sampling strategy. In order to verify the effect of the sampling strategy, we adjust

Just Accepted



(a) Mean normalized scores (b) Standard deviation

Fig. 7. Model prediction scores (mean normalized scores and standard deviation) under different video groups (divided by video length) on Kuaishou.



(a) Mean normalized scores (b) Standard deviation

Fig. 8. Model prediction scores (mean normalized scores and standard deviation) under different video groups (divided by video length) on Wechat.

Just Accepted

Table 8. Performance comparison *w.r.t.* $View_Time@K$ ($K = 3$) in different video groups on Kuaishou.

Group	1∼8	9∼18	19∼30	31∼40	41∼59
TReg	13.83	23.81	44.68	67.66	73.31
RReg	**22.95**	**35.30**	38.26	51.41	47.71
TRank	8.56	10.09	31.13	57.28	51.79
RRank	9.07	10.91	32.09	56.67	50.94
IPS	8.33	9.73	30.56	56.70	51.68
CausE	8.67	10.38	31.52	56.99	51.75
DecRS	9.51	31.73	**47.28**	**69.95**	**84.15**
DVR	14.08	26.28	43.20	65.71	72.62
VLDRec	10.29	14.78	35.85	58.16	54.41

Table 9. Performance comparison *w.r.t.* $View_Time@K$ ($K = 3$) in different video groups on Wechat.

Group	0∼13	14∼20	21∼30	31∼41	42∼59	60∼92	93∼120
TReg	6.60	15.90	28.45	45.31	49.98	88.62	125.17
RReg	**29.57**	**37.07**	**45.60**	49.76	59.34	64.81	70.20
TRank	9.46	18.34	30.74	42.97	40.24	68.38	111.88
RRank	9.28	17.52	30.64	42.09	42.29	69.17	112.52
IPS	9.68	17.75	28.45	37.48	34.60	61.74	106.93
CausE	12.35	22.72	33.84	45.00	43.26	66.72	107.96
DecRS	18.63	27.48	42.34	**55.77**	**74.75**	**107.21**	**139.93**
VLDRec	15.82	26.14	35.98	46.89	49.23	72.66	115.67

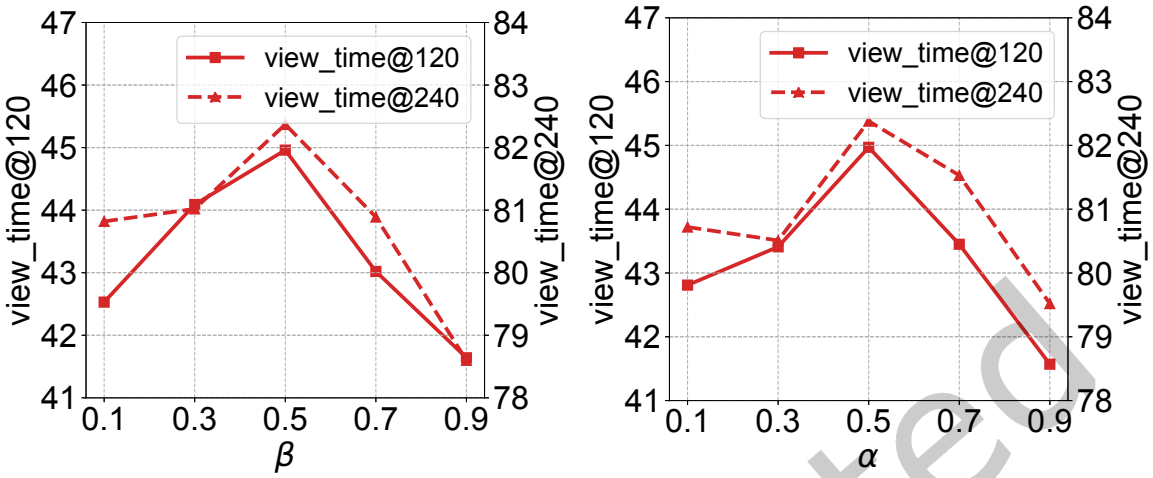
the value of β to observe changes in performance and show the result *w.r.t.* $View_Time@T$ in Fig. 9 (a). We can observe that the overall trend is rising first and then falling, which proves that the two sampling methods we proposed based on different objectives can effectively model the user's preferences and improve the performance of the model.

5.3.2 Performance on Different Weight of Multitask Learning. To alleviate the biased video length effect, VLDRec additionally learns to rank among positive and negative instances within the same video group and uses the multi-task learning strategy for training. To verify the effectiveness of this module, we adjust the hyper-parameter α of the multi-task learning strategy and illustrate the results in figure 9 (b). It can be observed that the performance of the model is the best when $\alpha = 0.5$. Model performance degrades when α increases or decreases, which means that the model has achieved a balance between the two goals of alleviating bias and fitting the data distribution, thereby improving the recommendation performance.

5.4 Study of Model Capability for Capturing User Preference (RQ3)

In terms of the model capability of learning user preferences on micro-videos, besides measuring the overall view time of users, another important dimension is to look at the micro-video content. In other words, the content of recommended micro-videos needs to match the user's interests. In this section, we analyze how exactly the recommendation results of different methods match the user's interests. Specifically, we use the video category

Just Accepted



(a) sampling strategy (b) multitask learning

Fig. 9. Performance with different weight of sampling strategy and multitask learning in Kuaishou.

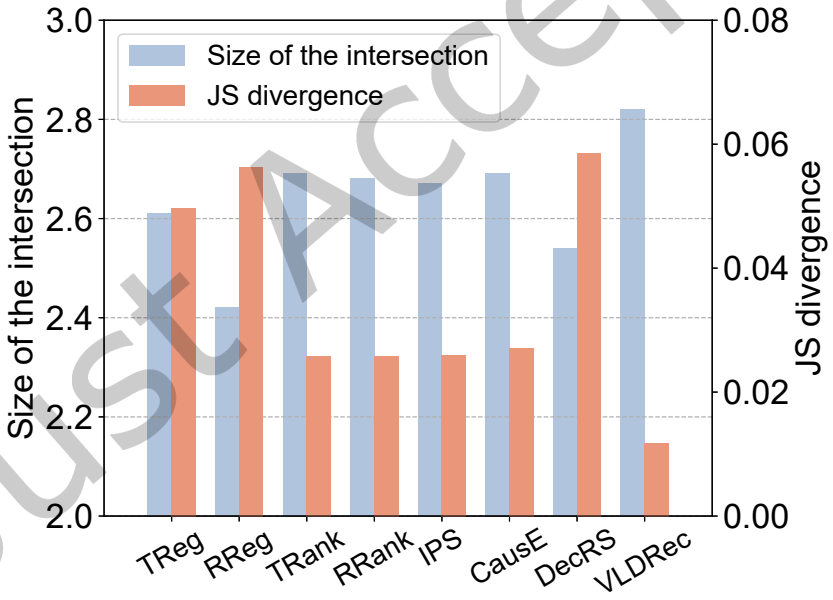


Fig. 10. The similarity between the recommendation results of different methods and the ground truth in Wechat.

information to verify the model's capability of capturing user interests, and this information is only available in Wechat dataset. Firstly, on an individual level, for each user, we compare the categories of the top five videos in the recommendation result with the categories of the five videos that the user actually watched the longest, and

Just Accepted

calculate the *size of intersection*. The larger the *size of intersection*, the recommendation results are closer to the user's preference. Results are shown in Fig. 10, where we can observe that the average *size of intersection* of our VLDRec method is the largest. The RateRegression method performs the worst, and the performance of other methods are also significantly worse than our proposed method. This means that the videos recommended by VLDRec are similar to the user's preference. Secondly, on a group level, for all users, we aggregate the top five videos recommended by the model and the videos that users actually watched the longest into two collections, respectively. To measure the similarity between these two micro-video distributions, we use the JSD metric. The smaller the JSD value, the higher the similarity between the two distributions. As shown in Fig. 10, we can observe that VLDRec has the smallest JSD value, while the regression method and DecRS perform poorly overall. In a word, VLDRec can effectively match the user's interests, and thus the recommendation results are more similar to the user's historical preference both at the individual and group level, while the common practices in many companies like regression based models only capture the biased preference that are strengthened by video length effect, as a result, its recommendation results cannot effectively match the interests of users.

6 CONCLUSION AND FUTURE WORK

In this paper, we aim to tackle the previously untouched problem of video-length effect in recommender systems for online micro-video platforms. We analyze the causes of the video-length effect and propose a VLDRec method for improving micro-video recommendation. By grouping the videos and designing the length-conditioned sampling method, we are able to generate unbiased pairs of training samples and learn the unbiased interests of users through a multi-task learning framework. Experimental results on both public and industrial datasets have proven the superiority of our method over previous solutions in terms of capturing real user preferences in collected user-video view data under a severe video-length bias.

Micro-video recommendation scenario differs from traditional scenarios from sample generation to evaluation due to the different user interfaces of online applications. In this work, we have made some explorations in above areas and we believe that, in the future, more in-depth studies are required, such as how to automatically define the preference labels in a smarter way. More importantly, since it is mainly the change of user interface that has spawned new problems of video-length effect, possible research works related to Computer–Human Interaction (CHI) seem necessary to expedite the problem resolution. Last but not least, building a new unbiased dataset with randomly exposed videos [13] can also help understand user behavior and make better recommendations.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China under grant 2020YFA0711403. This work is also supported by National Natural Science Foundation of China under 62272262, U22B2057, 62171260.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The connection between popularity bias, calibration, and fairness in recommendation. In *RecSys*. 726–731.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *SIGIR*. 385–394.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*. 104–112.

[4] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 11 (2013).

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. 335–344.

Just Accepted

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.

[8] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *ResSys*. 293–296.

[9] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *IJCAI*. 2230–2236.

[10] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.

[11] Jingtao Ding, Guanghui Yu, Yong Li, Xiangnan He, and Depeng Jin. 2020. Improving implicit recommender systems with auxiliary data. *ACM Transactions on Information Systems (TOIS)* 38, 1 (2020), 1–27.

[12] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *International Symposium on Information Theory (ISIT)*. IEEE, 31.

[13] 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 *CIKM*. 3953–3957.

[14] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[15] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal Inference in Recommender Systems: A Survey and Future Directions. *arXiv preprint arXiv:2208.12397* (2022).

[16] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline evaluation to make decisions about playlistrecommendation algorithms. In *WSDM*. 420–428.

[17] 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).

[18] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. 355–364.

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *WSDM*. 781–789.

[20] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations. In *KDD*. 3161–3171.

[21] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented fairness in recommendation. In *WWW*. 624–632.

[22] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in Recommendation: A Survey. *arXiv preprint arXiv:2205.13619* (2022).

[23] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *MM*. 1464–1472.

[24] Wei Lu, Fu-Lai Chung, Wenhao Jiang, Martin Ester, and Wei Liu. 2018. A deep Bayesian tensor-based system for video recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 1 (2018), 1–22.

[25] Yujie Lu, Yingxuan Huang, Shengyu Zhang, Wei Han, Hui Chen, Zhou Zhao, and Fei Wu. 2021. Multi-trends Enhanced Dynamic Micro-video Recommendation. *arXiv preprint arXiv:2110.03902* (2021).

[26] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *arXiv preprint arXiv:1910.14238* (2019).

[27] Elisa Mena-Maldonado, Rocío Cañamares, Pablo Castells, Yongli Ren, and Mark Sanderson. 2021. Popularity Bias in False-positive Metrics for Recommender Systems Evaluation. *ACM Transactions on Information Systems (TOIS)* 39, 3 (2021), 1–43.

[28] Aditya Pal, F Maxwell Harper, and Joseph A Konstan. 2012. Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Transactions on Information Systems (TOIS)* 30, 2 (2012), 1–28.

[29] Zhen Qin, Suming J Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-based propensity for unbiased learning in recommender systems: Algorithm and case studies. In *KDD*. 2359–2367.

[30] Yuhan Quan, Jingtao Ding, Chen Gao, Lingling Yi, Depeng Jin, and Yong Li. 2023. Robust Preference-Guided Denoising for Graph based Social Recommendation. In *Proceedings of the ACM Web Conference 2023*. 1097–1108.

[31] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[32] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*. PMLR, 1670–1679.

[33] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *CIKM*. 1161–1170.

[34] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *KDD*. 713–722.

[35] David Vallet, Frank Hopfgartner, Joemon M Jose, and Pablo Castells. 2011. Effects of usage-based feedback on video retrieval: a simulation-based study. *ACM Transactions on Information Systems (TOIS)* 29, 2 (2011), 1–32.

Just Accepted

[36] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank. In *CIKM*. 1475–1484.
[37] Jiayin Wang, Weizhi Ma, Jiayu Li, Hongyu Lu, Min Zhang, Biao Li, Yiqun Liu, Peng Jiang, and Shaoping Ma. 2022. Make Fairness More Fair: Fair Item Utility Estimation and Exposure Re-Distribution. In *KDD*. 1868–1877.
[38] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, Liqiang Nie, and Min Zhang. 2021. DualGNN: Dual Graph Neural Network for Multimedia Recommendation. *IEEE Transactions on Multimedia* (2021).
[39] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *KDD*. 1717–1725.
[40] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. 1288–1297.
[41] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.
[42] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *SIGIR*. 1001–1010.
[43] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *ICML*. PMLR, 6638–6647.
[44] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System. In *KDD*. 1791–1800.
[45] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *MM*. 3541–3549.
[46] Shuyuan Xu, Juntao Tan, Zuohui Fu, Jianchao Ji, Shelby Heinecke, and Yongfeng Zhang. 2022. Dynamic Causal Collaborative Filtering. In *CIKM*.
[47] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *CIKM*. 4321–4330.
[48] Zixuan Yi, Xi Wang, Iadh Ounis, and Craig Macdonald. 2022. Multi-modal Graph Contrastive Learning for Micro-video Recommendation. In *SIGIR*. 1807–1811.
[49] 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 *KDD*. 4472–4481.
[50] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*. 785–788.
[51] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*. 11–20.
[52] 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 *RecSys*. 43–51.
[53] Yu Zheng, Chen Gao, Jingtao Ding, Lingling Yi, Depeng Jin, Yong Li, and Meng Wang. 2022. DVR: Micro-Video Recommendation Optimizing Watch-Time-Gain under Duration Bias. In *Proceedings of the 30th ACM International Conference on Multimedia*. 334–345.
[54] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *WWW*. 2980–2991.

Just Accepted