Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation

Ruohan Zhan*
Hong Kong University of Science and Technology
Hong Kong SAR, China
ruohanzhan@gmail.com

Changhua Pei
Kuaishou Technology
Beijing, China
peichanghua@kuaishou.com

Qiang Su
Kuaishou Technology
Beijing, China
suqiang@kuaishou.com

Jianfeng Wen
Kuaishou Technology
Beijing, China
wenjianfeng@kuaishou.com

Xueliang Wang
Kuaishou Technology
Beijing, China
wangxueliang03@kuaishou.com

Guanyu Mu
Kuaishou Technology
Beijing, China
muguanyu@kuaishou.com

Dong Zheng
Kuaishou Technology
Beijing, China
zhengdong@kuaishou.com

Peng Jiang
Kuaishou Technology
Beijing, China
jiangpeng@kuaishou.com

ABSTRACT

Watch-time prediction remains to be a key factor in reinforcing user engagement via video recommendations. It has become increasingly important given the ever-growing popularity of online videos. However, prediction of watch time not only depends on the match between the user and the video but is often mislead by the *duration* of the video itself. With the goal of improving watch time, recommendation is always biased towards videos with long duration. Models trained on this imbalanced data face the risk of bias amplification, which misguides platforms to over-recommend videos with long duration but overlook the underlying user interests.

This paper presents the first work to study duration bias in watch-time prediction for video recommendation. We employ a *causal graph* illuminating that duration is a *confounding* factor that concurrently affects video exposure and watch-time prediction—the first effect on video causes the bias issue and should be eliminated, while the second effect on watch time originates from video intrinsic characteristics and should be preserved. To remove the undesired bias but leverage the natural effect, we propose a **D**uration-**D**econfounded **Q**uantile-based (**D2Q**) watch-time prediction framework, which allows for scalability to perform on industry production systems. Through extensive offline evaluation and live experiments, we showcase the effectiveness of this duration-deconfounding framework by significantly outperforming the state-of-the-art baselines. We have fully launched our approach

on Kuaishou App, which has substantially improved real-time video consumption due to more accurate watch-time predictions.

CCS CONCEPTS

KEYWORDS

1 INTRODUCTION

The rise of online video consumption has drawn growing efforts to optimize recommender systems of internet-based Video on Demand (VOD) systems such as YouTube and streaming players such as TikTok, Instagram Reels, and Kuaishou (see demonstration in Figure 1). As such, a main goal is to improve the amount of time users spend on watching the videos, referred to as expected *watch time* [7]. Watch time is a dense signal existing in each video view that concerns every user and video on the platform and represents a scarce resource of user attention that companies compete for. It is thus crucial to accurately estimate watch time on candidate videos when a user arrives. Accurate predictions enable the platform to recommend videos with potentially large watch time to improve user engagement, which directly drives the critical production metric of daily active user (DAU) and thereby the revenue growth.

Watch time is mainly affected by two factors. As known, it is largely governed by how interested the user is in the video and

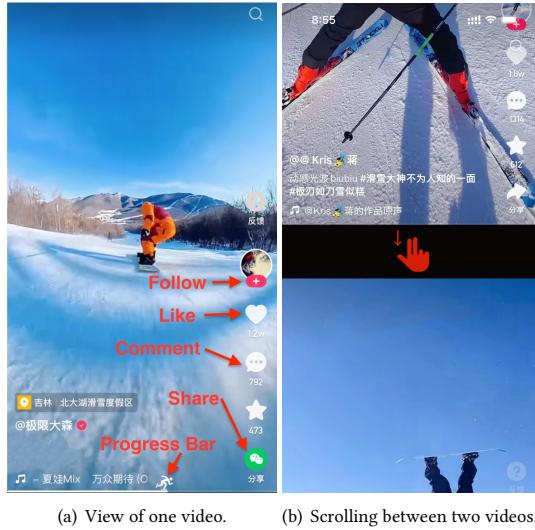(a) View of one video. (b) Scrolling between two videos.

Figure 1: A screenshot of the video recommendation module on Kuaishou App. Videos are displayed in the full-screen manner for immersive user experience.
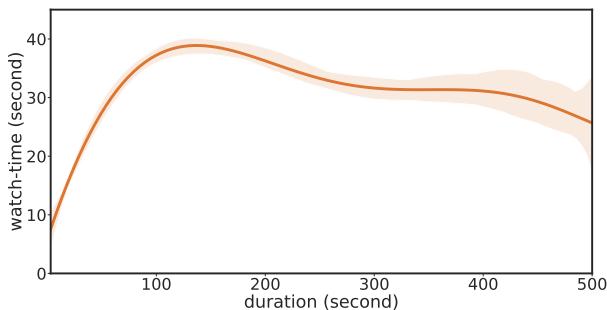

Figure 2: 60th percentile watch-time on videos with respect to duration. Data is collected from Kuaishou App with sample size surpassing 20 billion. The spanned area denotes the 99.99% confidence interval of watch time.

can be zero when there is no interest match at all [12, 29]. Meanwhile, *duration* of a video itself (*i.e.*, length of the video) also plays a significant role in determining how long the user spends on the video. Figure 2 shows that user watch-time is positively correlated with video duration. As a result, standard watch-time prediction models often use duration together with other video characteristics as feature inputs to make predictions [7, 8]. However, such practice unfortunately results in a bias issue in many recommender systems. Figure 3 demonstrates that recommendation is over time progressively based towards videos with long duration, due to the platform's goal of maximizing user watch time. As a result, videos with longer duration are likely to be over-exposed, such that user real interest is undervalued in recommendations. More severely, models trained on such imbalanced data will amplify the duration

bias due to the system's feedback loop [27], which undesirably harms the diversity and personalization in ideal recommendation.

Despite the high prevalence, duration bias is much less explored as compared to many other biases that are caused by item popularity or position in recommendation studies [1, 2, 18, 31, 38–40]. With the goal of maximizing user watch-time, recommender systems may learn spurious correlation between duration and watch-time; thus videos with longer duration are more likely to be shown even though they may fail to match the user interest well. On the other hand, videos with long duration usually have larger sample size resulted from existing imbalanced exposure, which may dominate model learning such that model performance varies across duration.

This paper presents the first work in studying duration bias in watch-time prediction. We employ a direct acyclic graph (named as *causal graph* [20]) to characterize the causal relationship regarding duration in watch-time prediction, modeled by Figure 4(a). Specifically, duration—served as a *confounding* factor [20]—simultaneously affects both watch-time prediction and video exposure. The first effect of duration on watch-time shows that users tend to spend more time watching videos with intrinsically longer duration, which is a natural effect and should be captured by watch-time prediction models. The second effect from duration to video, however, is a bias term that plagues many watch-time prediction models. Such effect suggests that duration influences the likelihood of video impression, which represents model's unfair preference on videos with long duration and should be eliminated. Such explicit modeling of duration effects, in contrast to previous works that only use duration as features for watch-time prediction, allows us to remove the undesired bias but preserve the true influence.

To deal with the duration bias, we follow the principle of *backdoor adjustment* [21] and intervene the causal graph of watch-time prediction to remove the undesired effect from duration to video exposure, as characterized in Figure 4(b). We note that the effect from duration to watch-time is preserved, since such relationship is intrinsic and should be leveraged in prediction. Operationally, we split the training data into equal parts with respect to duration; and for each duration group, we learn a regression model to predict group-wise watch-time quantiles, where the labels are determined by the original watch-time values and the empirical cumulative distribution of watch time in the corresponding group. Such quantile prediction enables model parameter sharing across duration groups, bringing in benefits on scalability. Together, we summarize our contributions as below:

- **Causal Formulation of Duration Bias in Watch-time Prediction.** We employ a causal graph to formalize the overlooked but widely existing issue of duration bias in watch-time prediction. We point out that duration is a confounder that affects both watch-time prediction and video exposure, of which the former is intrinsic and should be preserved, while the latter is bias and should be eliminated.
- **Adjusting for Duration with Scalability.** Guided by backdoor adjustment, we split data based on duration and fit a watch-time prediction model for each duration group to remove the duration bias on video exposure. We modify the watch-time labels with regard to duration to allow for parameter sharing across groups and gain scalability.
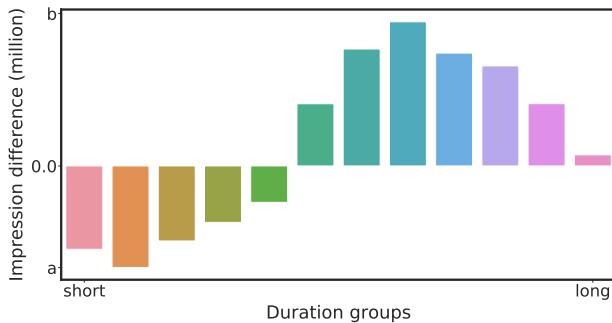
Figure 3: Change of video impression over 11 months on Kuaishou App per video duration. Bins are sorted in the ascending order of duration. The height of bin represents the difference of impression counts over this period. The absolute values have been omitted for confidentiality purposes. With the platform's goal to improve watch time, impression is progressively biased towards videos with long duration.

- **Extensive Offline Evaluation.** We conduct a series of offline evaluation on data collected from Kuaishou App to demonstrate the advantages of our model over existing baselines. We further do ablation studies on the number of duration groups, enlightening that with group number increasing, our model performance firstly improves (thanks to duration de-biasing) and then declines (due to increased estimation error from reduced groupwise sample size).

- **Benefits in Live Experiments.** We further implement our approach in live experiments to facilitate video recommendations on Kuaishou platform, showing that by removing the undesired duration bias, our approach improves watch-time prediction accuracy and contributes to optimized real-time video consumption as compared to existing strategies.

2 RELATED WORK

Watch-time Prediction. It is crucial for many industry recommender systems to accurately predict watch time, which is one of the most representative metrics of user engagement [7, 28, 36]. However, in contrast to other metrics such as Click-Through-Rate (CTR) [5, 22, 24, 25, 29, 41], there are fewer research endeavors focused on this area. Remarkably, [7] provides one of the industry standard solutions to watch-time prediction, where the regression problem is transformed into Weighted Logistic Regression (which shall be referred to as WLR for the remaining of this paper), and the impressed videos are weighted with the actual watch time; in this way, the learned odds are approximately the expected watch time (detailed derivation is shown in Appendix A). However, videos with long duration—which has positive correlation with watch time as shown in Figure 2—often get larger sample weights during model training, amplifying the duration bias. Beside, such approach cannot be directly adapted to streaming services such as TikTok and Kuaishou that provide full-screen video content for immersive user experience, where there are no nominal unimpressed/unclicked samples—every video sample has been shown to a user and is thus impressed. In this paper, we revise the method in [7] to adapt

to both streaming services (TikTok and Kuaishou) and scenarios as YouTube with user behaviors of click and watch. The modified method (*i.e.,* WLR) acts as one of baselines in experiment Section 5.1 and Section 5.2 .

Bias in recommender systems. Our work is also closely related to a growing literature focused on addressing biases in recommender systems [4]. One line of such strives to mitigate systematic biases that stand in opposite to fairness [35] and social welfare [10], via promoting equity of attention for items or improving model performances for subgroups [1, 2, 18, 38]. Another line focuses on dealing with algorithmic biases to improve model performance and break bias amplification resulted from feedback loop, which we view our work as complementary to. Methods in this line can be broadly categorized into three classes: (i) *causal embedding*, where researchers decompose item embedding with respect to different causal effects and learn each embedding using a curated bias-free dataset associated with the corresponding effect [3, 13, 40]; (ii) *inverse propensity weighting*, where researchers reweights samples following the rule of importance sampling [6, 16, 26] to correct for the distributional shift in training data, such that the learned model captures the bias-free signals to generalize to unseen distributions [11, 26, 32]; and (iii) *causal intervention*, where researchers intervene the causal relationship that causes the bias and add adjustment to eliminate this harmful effect for more accurate estimation [31, 37, 39]. Our work is classified into the third category, where with causal intervention, we remove the undesired effect of duration on video but preserve the desired effect of duration on watch time.

Causal Intervention. We finally review a relevant line in causality-related observational studies [23]. The causal relationships among variables are captured by a directed acyclic graph, named as *causal graph*, where nodes denote variables and directional edges denote causal effects [20]. Bias often arises when a model fails to account for a variable—referred to as *confounder*—which simultaneously affects both the feature variables and the outcome [19, 20]. In our case, this variable is the duration that has the confounding effects on both video and watch time. To deal with the bias and deconfound the problematic variable, a standard approach is to conduct *do-calculus* that specifies the value of relevant variables, named as *backdoor adjustment*, such that the intervened causal graph eliminates the edge of undesired causal effect [21]. Such approach has been widely used to estimate causal effect in various domains across healthcare [14], bioinformatics [17], and socioeconomics [30]. More recently, in recommender systems, it receives growing applications to adjust for the item popularity bias for CTR prediction [31, 39].

3 CAUSAL MODEL OF WATCH-TIME PREDICTION

Our goal is to predict watch time of a user when she/he is recommended with a video. We start by formulating the problem via a causal graph that characterizes the relationships among user, video, duration, watch-time, and in particular, the confounding effects of duration on both video exposure and watch-time prediction in most recommender systems, as shown in Figure 4(a):

- *U* denotes user representation, including user demographics, instantaneous context, historical interactions, etc.
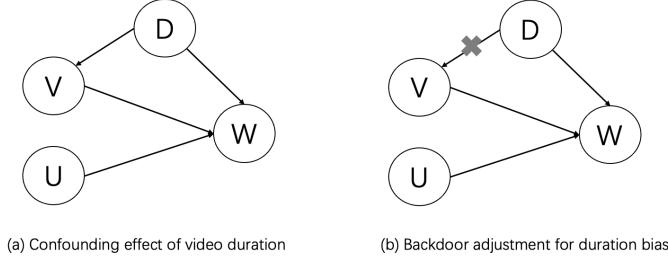
(a) Confounding effect of video duration   (b) Backdoor adjustment for duration bias

Figure 4: Causal graphs of watch-time prediction: U–user, V–video, D–duration, W–watch time. **Figure (a) models the confounding effect of video duration on both video exposure and watch-time prediction. Figure (b) uses backdoor adjustment to deconfound duration and remove its effect on video.**

- V denotes video representation, including video topics, etc.
- D denotes video duration, i.e., the length of the video.
- W denotes the time user spends on watching the video.
- $\{U, V\} \to W$ capture the *interest* effect on watch-time, which measures how user is interested in the video.
- $D \to W$ captures the *duration* effect on watch time, which suggests that when two videos match user interest comparably, longer video may receive longer watch time.
- $D \to V$ implies that duration will affect video exposure. Recommender systems often have unfair preference on videos with long duration, due to the goal of maximizing user watch time; such bias is amplified by feedback loop, as shown by Figure 3. Besides, duration may affect mode training since (i) sample size varies with duration—videos with long duration usually have larger sample size, on which the prediction model has better performance; and (ii) videos with different duration may receive different sample weights in standard models such as WLR [7], which affects the allocation of gradient in model training.

Notably, the causal graph in Figure 4(a) demonstrates that duration is a confounder that affects watch-time via two paths: $D \to W$ and $D \to V \to W$. The first path suggests that duration has a direct causal relationship with watch time, which should be captured by watch-time prediction models since users tend to spend more time on watching long videos versus short ones. However, the second path implies that video exposure is undesirably affected by its duration, and thus the video distribution is biased towards long videos; and if not eliminated, predictions would face the risk of bias amplification by the feedback loop of recommender systems.

4 BACKDOOR ADJUSTMENT FOR DURATION BIAS

In this section, we follow the principle of backdoor adjustment to deconfound duration, where we remove the bias from duration on video but preserve the effect from duration on watch time. We propose a scalable watch-time prediction framework that is Duration-Deconfounded and Quantile-based (D2Q), manifested by

   i. Spitting data based on duration to remove duration bias;

  ii. Fitting watch-time quantiles instead of the original values to enable parameter sharing across groups for scalability.

We summarize our training and inference procedures in Algorithms 1 and 2 respectively.

Algorithm 1: Training of D2Q: Duration-Deconfounded Quantile-based Watch Time Prediction

Input: training data $\{(u_i, v_i, d_i, w_i)\}_{i=1}^n$.
  (1) Compute empirical quantiles of duration $\{d_i\}_{i=1}^n$ to determine the duration groups $\{\mathcal{D}_k\}_{k=1}^M$.
  (2) Split data $\{(u_i, v_i, d_i, w_i)\}_{i=1}^n$ into M equal parts based on $\{\mathcal{D}_k\}_{k=1}^M$.
  (3) For each duration group $\mathcal{D}_k$, compute the empirical cumulative distribution of watch time $\widehat{\Phi}_k$ on data $\{w_i : d_i \in \mathcal{D}_k\}$.
  (4) Solve the watch-time quantile prediction model h using all samples:

$$h = \arg\min_{h'} \sum_{\{(u_i, v_i, w_i)\}_{i=1}^n} \left(h'(u_i, v_i) - \widehat{\Phi}_{k_i}(w_i)\right)^2,$$

    where k_i is the duration group of sample i such that $d_i \in \mathcal{D}_{k_i}$.

Output: duration groups $\{\mathcal{D}_k\}_{k=1}^M$, watch-time quantile prediction model h.

Algorithm 2: Inference of D2Q: Duration-Deconfounded Quantile-based Watch Time Prediction

Input: user-video pair (u_0, v_0) to be inquired, duration groups $\{\mathcal{D}_k\}_{k=1}^M$, watch-time quantile prediction model h.
  (1) Find the corresponding duration group $\mathcal{D}_{k_0}$ for video v_0.
  (2) Estimate watch time $\widehat{w}_0 = \widehat{\Phi}_{k_0}^{-1}\left(h(u_0, v_0)\right)$.

Output: estimated watch-time $\widehat{w}_0$.

4.1 Deconfounding Duration

Following the *do*-calculus, we block the duration effect on video exposure by removing edge $D \to V$, as illustrated by the deconfounded causal graph G_1 in Figure 4(b). We frame the watch-time prediction model as $\mathbb{E}[W|do(U,V)]$ and have

$$\mathbb{E}[W|do(U,V)] = \mathbb{E}_{G_1}[W|U,V]$$

$$\stackrel{(i)}{=} \sum_d \mathbb{P}_{G_1}(D=d|U,V)\,\mathbb{E}_{G_1}[W|U,V,D=d]$$

$$\stackrel{(ii)}{=} \sum_d \mathbb{P}_{G_1}(D=d)\,\mathbb{E}_{G_1}[W|U,V,D=d] \qquad (1)$$

$$\stackrel{(iii)}{=} \sum_d \mathbb{P}(D=d)\,\mathbb{E}[W|U,V,D=d],$$

where (i) is by law of total expectation [33]; (ii) is because D is independent of $\{U, V\}$ with the intervention that removes edge $D \to V$ in graph G_1; and (iii) is because such intervention does not

change the distribution of W conditioning on $\{U, V, D\}$, and the marginal distribution of D remains the same.

Equation (1) sheds light on the design to deconfound duration: one can estimate $\mathbb{P}(D)$ and $\mathbb{E}[W|U, V, D]$ separately and then combine them together to construct the final estimation. In this paper, we propose to discretize the duration distribution $\mathbb{P}(D)$ into disjoint groups and fit group-wise watch-time prediction model $\mathbb{E}[W|U, V, D]$ to complete the estimation.

4.2 Data-Splitting based on Duration Quantiles

We now present a general framework to estimate the watch-time with duration deconfounded, depicted by Figure 4(b). The high-level idea is to split data based on duration and construct group-wise watch-time estimation to debiase duration on video exposure.

Specifically, to block edge $D \rightarrow V$, we split training samples into M equal parts based on duration quantiles, which discretizes the distribution $\mathbb{P}(D)$ into disjoint components. Let $\{\mathcal{D}_k\}_{k=1}^M$ be these duration groups. Continuing the derivation in (1), we estimate the deconfounded model $\mathbb{E}[W|do(U, V)]$ via the approximation:

$$
\begin{aligned}
\mathbb{E}[W|do(U, V)] &= \sum_d \mathbb{P}(D = d)\,\mathbb{E}[W|U, V, D = d] \\
&\approx \sum_{k=1}^M \mathbf{1}\{d \in \mathcal{D}_k\}\,\mathbb{E}[W|U, V, D \in \mathcal{D}_k] \\
&\triangleq \sum_{k=1}^M \mathbf{1}\{d \in \mathcal{D}_k\} f_k(U, V),
\end{aligned} \tag{2}
$$

where for each duration group $\mathcal{D}_k$, $f_k(u, v)$ is the watch-time prediction model fitted on samples $\{(u_i, v_i, w_i, d_i) : d_i \in \mathcal{D}_k\}$ that belong to the duration group.

We hereby provide an intuitive explanation on why such data splitting procedure based on duration can mitigate the bias issue of edge $D \rightarrow V$ in Figure 4(a). In standard watch-time prediction models such as WLR [7], samples with long watch-time weights more in the gradient updating, such that the prediction model often performs poorly on samples with short watch-time. Watch-time is highly correlated with duration, as shown in Figure 2. By splitting data based on duration and fitting model group-wisely, we alleviate the interference from samples with long watch time on samples with short watch time during model training.

However, such data-splitting approach raises another concern. If we fit an individual watch-time prediction model f_k for each duration group $\mathcal{D}_k$ (as demonstrated in Figure 5(a)), model size will grow undesirably large, which is not practical in real production systems for scalability concern. But if we allow parameter sharing across duration groups, fitting with the original watch-time labels is equivalent to learning without data splitting, which fails to bring in the benefits of duration deconfounding. The following section explains how to address this dilemma via transforming the original watch-time labels into duration-dependent watch-time labels, allowing us to both remove duration bias and also maintain a single set of model parameters to gain scalability.

4.3 Estimating Watch-time per Duration Group

Moving on, we describe how we fit a single watch-time prediction model using data from all duration groups. Recall that the goal of

our design is twofold: (i) duration debiasing and (ii) parameter sharing. The key is to transform the watch-time label to be duration-dependent, manifested by fitting watch-time quantiles—instead of the original values—with respect to the corresponding duration group. Collectively, we introduce our **D**uration-**D**econfounded **Q**uantile-based (D2Q) watch time prediction framework.

Denote $\widehat{\Phi}_k(w)$ as the empirical cumulative distribution of watch time on videos in duration group $\mathcal{D}_k$. Given a user-video pair (u, v), the D2Q method predicts its watch-time quantile in the corresponding duration group and then maps it to the value domain of watch time using $\widehat{\Phi}_k$. That is,

$$
f_k(u, v) = \widehat{\Phi}_k^{-1}(h(u, v)), \tag{3}
$$

where h is a watch-time quantile prediction model fitted on data across *all* duration groups:

$$
h = \arg\min_{h'} \sum_{\{(u_i, v_i, w_i)\}_{i=1}^n} \left(h'(u_i, v_i) - \widehat{\Phi}_{k_i}(w_i)\right)^2, \tag{4}
$$

with k_i being the duration group of sample i such that $d_i \in \mathcal{D}_{k_i}$. One can apply any off-the-shelf regression model to fit the quantile prediction model h and maintain one single set of model parameters across all duration groups. Then, during the inference phase, when a new user-video pair (u_0, v_0) arrives, the model firstly finds the duration group $\mathcal{D}_{k_0}$ that video v_0 belongs to, and then maps the watch-time quantile prediction $h(u_0, v_0)$ to the watch-time value $\widehat{\Phi}_{k_0}^{-1}(h(u_0, v_0))$. We summarize the learning and inference procedures in Algorithm 1 and 2 respectively.

In this way, D2Q fit labels that are duration-dependent. We note that video duration should also be part of model input to differentiate samples from different duration groups, as illustrated by Figure 5(b). Otherwise, samples from different duration groups may share the same label of watch-time quantile but have different characteristics—a single model would fail to learn the watch-time quantile across groups. To fully utilize duration information, one can additional incorporate a duration adjustment tower in the model architecture as ResNet [15], for which we refer to as Res-D2Q and illustrate in Figure 5(c). Section 5 demonstrates that Res-D2Q further improves watch-time prediction accuracy over D2Q.

The transformation of watch-time labels with respect to duration allows for both deconfounding duration bias and parameter sharing across duration groups. However, with the number of duration groups increasing, the group sample size shrinks, and the empirical cumulative distribution of watch-time per duration group will gradually deviate from its true distribution. Therefore, the model performance should firstly improve with duration-based data splitting, thanks to the benefits of deconfounding duration; then with the number of duration groups growing, the estimation error of empirical watch-time distribution will dominate the model performance and make it progressively worse. Section 5 empirically justifies this performance change with a series of experiments.

5 EXPERIMENTAL RESULTS

In this section, we provide empirical evidence to demonstrate the effectiveness of our approach on both real-world data and live experiments. Extensive offline evaluation shows that our method outperforms existing baselines by providing more accurate watch-time
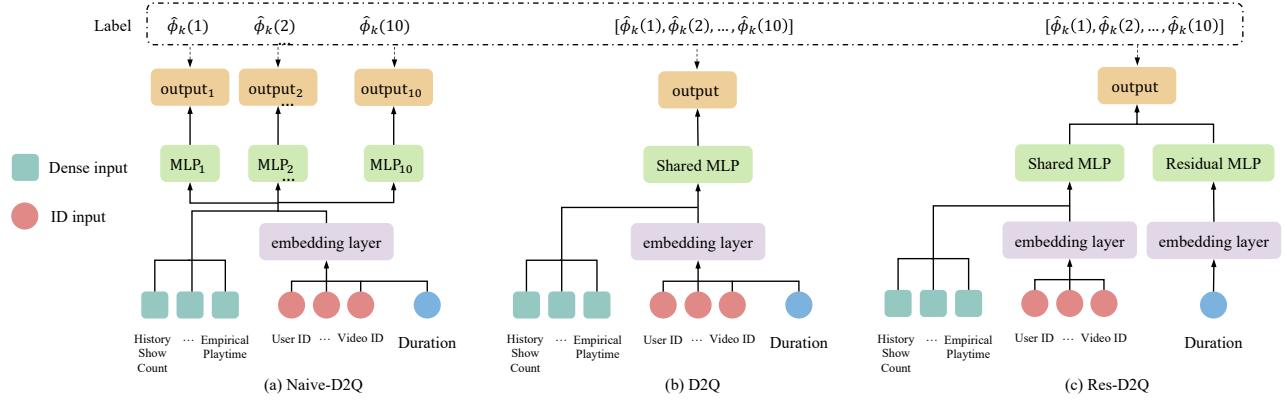
Figure 5: Different model architectures for estimating watch-time per duration group, *i.e.,* $\widehat{\Phi}_k(h(u,v))$. Figure (a) fits individual models to estimate watch-time per duration group. "Dense input" refers to historical statistical numbers (such as *historical show count, empirical watchtime*) of the video. "ID input" refers to ID features (such as *user id, video id*) and categorical features (such as *video category, user gender*). Figure (b) fits a single model across all duration groups, with labels of watchtime quantiles calculated via watch-time empirical distribution in the corresponding duration group. Figure (c) further utilizes duration information in the network architecture and consequently improves watch-time estimation.

prediction, such that the ranking order induced by the predicted values is closer to the ideal ranking. We note that with the platform's goal of improving user watch time, ranking—in contrast to the true watch-time value—is usually much valued when recommending videos in practice. Furthermore, by incorporating our method in the recommender system of a short video platform, we find it effectively improves real-time video consumption as compared to alternatives, thanks to its ability to generate better ranking of candidate videos based on optimized watch-time prediction.

5.1 Offline Evaluation

We first evaluate our approach as well as other baselines on offline data collected from real applications [1]. In particular, we are interested in understanding: (i) *how does deconfounding duration contribute to watch-time prediction?*; and (ii) *how does the number of duration groups affect our model performances?*

5.1.1 Data. We use production data collected from online recommender systems on the Kuaishou App. By the nature of full-screen feed recommendation, every video in the collected sample has been shown to a user and is associated with the user watch time (which can be close to zero if the user immediately scrolls down to the next video). Specifically, for the causal graph shown in Figure 4, we have

- user representation U: user instantaneous context (such as location, time, and device), stationary context (such as demographics if available), and historical interactions that encode his/her interests.
- video representation V: video topic information, its corresponding video-creator information, and its previous interactions with other users.
- duration D: length of the video.
- watch-time W: the watch time from the user.

[1]Reproduction code can be found at https://github.com/MorganSQ/Ks-D2Q

Table 1: Summary statistics of Kuaishou dataset used in offline evaluation in Section 5.1

Users	Photos	Categories	Instances
47,298,353	15,187,170	5,942	1,346,539,657

All algorithms evaluated share the same input features. In total, we have 1,211,885,691 samples for training and 134,653,965 samples for testing, with statistics summarized in Table 1.

5.1.2 Methods. We focus on the following methods.

- **VR (Value Regression).** This approach predicts the watch-time value directly, via minimizing mean squared error loss between the predicted value and the actual watch time.
- **WLR (Weighted Logistic Regression) [7].** This approach fits a weighted logistic regression model and uses the learned odds as the predicted watch time. Since there are no unimpressed videos in our case, we determine the binary labels based on whether the watch time surpasses the q_{60}-quantile of the empirical watch-time distribution, which is computed on all training samples. Following [7], positive samples are weighted by watch-time and negative samples receive unit weight. Appendix A details this method.
- **D2Q (Ours).** As described in Section 4.3, this approach (i) splits data based on duration; and (ii) fits a regression model—with its architecture shown in Figure 5(b)—to estimate watch time quantile via mean squared error loss. Then the predicted quantile is mapped to the watch-time value domain—based on the group-wise empirical watch-time distribution—to output the final watch-time estimation.
- **Res-D2Q (Ours).** This method further utilizes the duration information by improving D2Q and incorporating duration

Table 2: Offline evaluation results on Kuaishou dataset. We use bold fonts to label best performances.

#Groups	Method	Kuaishou dataset.		
		XAUC	XGAUC	MAE
1	VR	0.6843	0.6380	28.2413
	WLR[7]	0.6940	0.6436	65.6535
	D2Q	0.7084	0.6562	36.4871
10	D2Q	0.7092	0.6579	26.8684
	Res-D2Q	0.7108	0.6604	26.6686
20	D2Q	0.7147	0.6654	26.5993
	Res-D2Q	0.7150	0.6679	26.5530
30	D2Q	0.7148	0.6660	26.5227
	Res-D2Q	**0.7145**	**0.6693**	**26.5073**
50	D2Q	0.7142	0.6619	26.8047
	Res-D2Q	0.7141	0.6643	27.1260
100	D2Q	0.7119	0.6608	26.8277
	Res-D2Q	0.7125	0.6637	26.7297

in model network layers, following the design of ResNet. The model architecture is illustrated in Figure 5(c).

All algorithms share the same network architecture, except that for the classification-based algorithm WLR and quantile-prediction algorithms D2Q and Res-D2Q, we rescale the output via a sigmoid function to be within $[0, 1]$; and for Res-D2Q, we add a residual multi-layer perception (MLP) for duration adjustment in the last layer to help the model differentiate samples from different duration groups. Appendix B specifies the detail of network architecture. For both of our duration-deconfounded algorithms D2Q and Res-D2Q, we vary the number of duration groups across $[1, 10, 20, 30, 50, 100]$ to study its influence on model performance.

5.1.3 Metrics. We consider the following metrics for performances.

- **MAE (Mean Absolute Error)**, which measures the mean absolute error between the predicted and true values,

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{5}$$

where $y_i, \hat{y}_i$ are the true and predicted values for sample i.
- **XAUC**, which is an extension of AUC to dense values. For a pair of samples, we score 1 if the predicted watch-time values of the two videos are in the same order as the ground truth and score 0 vice versa. We uniformly sample such pairs from test set and average those scores as XAUC. Intuitively, XAUC measures how the ranking induced by the predicted watch time is in agreement with the ideal ranking. Larger XAUC suggests better model performance.
- **XGAUC**, which calculates XAUC per user and then averages scores with weights proportional to user sample size. Larger XGAUC suggests better model performance.

We note that the ranking-order-related metrics such as XAUC and XGAUC are often valued more in real applications than the absolute value accuracy measured by MAE, since platforms generate recommendations based on the ranking of predicted values.
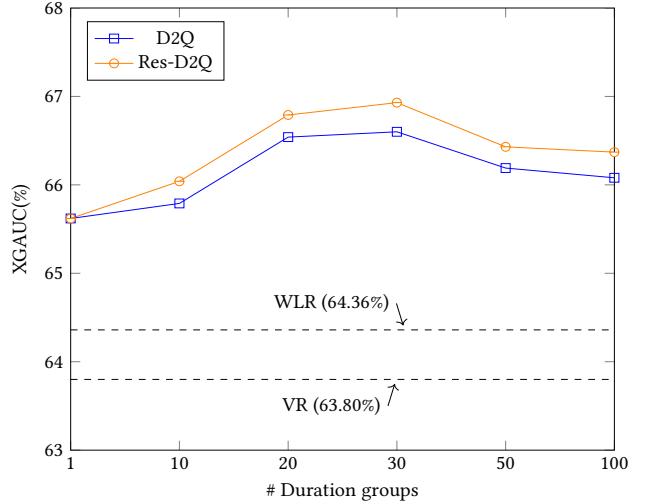

Figure 6: XGAUC of watch-time prediction methods with respect to number of duration groups. Our models D2Q and Res-D2Q significantly outperform the baseline methods WLR and VR. Our model performance first improves gradually with increased number of duration groups, benefiting from deconfounding duration. Then the performance drops when the group number is too large, due to reduced group-wise sample size and accumulated estimation error.

5.1.4 Result-I: Overall performance. Table 2 shows the performances of different methods with respect to various number of duration groups. Note that there is no data splitting for VR and WLR, and thus we present their results in the row of group number equaling one; when there is only one group, Res-D2Q is equivalent to D2Q since all samples share the same duration adjustment, and thus we omit the result of Res-D2Q there.

Our approaches D2Q and Res-D2Q with 30 duration groups peak the performance in all metrics XAUC, XGAUC, and MAE. In particular, by further leveraging the duration information in the model architecture, Res-D2Q can better distinguish samples from different duration groups and thus outperforms D2Q in most cases. When there is no data splitting, D2Q (which fits the watch time quantile directly) has comparable performance with LR (which fits the watch time value). However, once data is split based on duration, D2Q with any experimented number of duration groups generates more accurate prediction than LR, endorsing the effectiveness of our duration-wise data splitting to deconfound duration.

5.1.5 Result-II: Effect of number of duration groups. Figure 6 plots the XGAUC of our methods D2Q and Res-D2Q with respect to number of duration groups. When there is no data splitting, both methods are equivalent to each other. Once data is split to deconfound duration, Res-D2Q is superior to D2Q via improved network architecture with duration information. With the number of duration groups increasing, the performance firstly improves, with the merit of deconfounding duration by data splitting, and then

Table 3: Live experiments on Kuaishou App. We use VR as a baseline and show the relative performance of WLR and Res-D2Q with #*Groups* = 30. The square brackets represent the 95% **confidence intervals for online metrics. Statistically-significant improvement (whose value is not in the confidence interval) is marked with bold font in the table.**

Method	Main Metric.	Constraint Metrics.			
	Watch Time	Like	Follow	Share	Comment
WLR *v.s.* VR (baseline)	**+0.184**%	**+1.012**%	+0.214%	+0.959%	-0.137%
	$[-0.16\%, 0.16\%]$	$[-0.50\%, 0.51\%]$	$[-0.4\%, 0.4\%]$	$[-1.31\%, 1.40\%]$	$[-0.75\%, 0.73\%]$
Res-D2Q *v.s.* VR (baseline)	**+0.746**%	+0.251%	-0.167%	-0.861%	+0.271%
	$[-0.15\%, 0.15\%]$	$[-0.41\%, 0.41\%]$	$[-0.6\%, 0.6\%]$	$[-1.21\%, 1.21\%]$	$[-0.85\%, 0.86\%]$

declines, due to the increased estimation error of empirical watch-time distribution resulted from the shrinkage of sample size. Such observation is in agreement with the discussion in Section 4.3.

5.2 Live Experiments

We further test our approach in live A/B experiments powered by Kuaishou video recommendation platform, demonstrating its advantage over alternatives in improving real-time video consumption, as a consequence of improved watch-time prediction.

5.2.1 Compared methods. We compare baseline methods VR and WLR with our approach Res-D2Q with 30 duration groups, which achieves the best performance in offline evaluation in Section 5.1. Detailed description of methods can be found in Section 5.1.2. We use the same model architecture as that in Section 5.1.

5.2.2 Experimental setup. We integrate watch-time prediction into the ranking phase of the online recommender system used by Kuaishou App. Specifically, when a user arrives, the recommender system first generates a set of candidate videos that the user might be interested in, based on his/her characteristics. Then, the prediction model predicts the watch time of each video candidate supposing that it were recommended to the user. The candidate videos are ranked in accordance with the predicted values, and videos with higher values have larger likelihood to be recommended.

Models are pretrained on shared samples collected within one day before being tested in real time. We conduct A/B experiments to evaluate their live performances—we randomly split users on the platform into buckets of 5%, 5%, 5%, 85% and use the first three buckets for online evaluation, *i.e.*, 5% of the users on Kuaishou App experience the recommender system that embodies the corresponding watch-time prediction model. Considering that Kuaishou serves over 320 million daily active users [9], doing experiments in the 5%-bucket affects a huge population of more than 15 million users, which endorses the statistical significance of our results.

5.2.3 Metrics. We evaluate model performance based on the total amount of time spent by users in the bucket on watching the videos, denoted as WatchTime. This metric is positively correlated with watch-time prediction accuracy. Consider a mental thought: there are two watch-time prediction methods model-A and model-B, and suppose that model-A better predicts watch time. Then videos with larger ground-truth watch time are more likely to be ranked higher and consequently shown to users by the recommender system with model-A, as compared to that with model-B; as a result, users in

the bucket that tests model-A will spend more time on watching the videos, improving the WatchTime metric.

We also provide additional metrics widely-adopted in real production systems, which measure user engagement from the perspective of user interactions that are also valued by platforms, including *like* (clicking like button on the screen), *follow* (following the corresponding video-creator), *share* (sharing the video with friends), and *comment* (leaving comments on the video). However, since interactions are not mutually exclusive among platforms (contrary to WatchTime), platforms usually use them as constraint metrics when evaluating strategies that improve WatchTime. We present the total number of like/follow/share/comment counts for each method to complement our evaluation.

5.2.4 Results. Table 3 shows the performance of evaluated methods after being tested online concurrently for 24 hours on Kuaishou App. We use VR as a base method and list the relative performances of WLR and Res-D2Q. For the main metric WatchTime, both WLR and Res-D2Q outperform VR with statistical significance, and our method Res-D2Q achieves a larger improvement of +0.746%, which is remarkable given the fact that the average watch-time improvement from production algorithms is around 0.1%. Indeed, our approach has been deployed to the online recommender system on Kuaishou App. For the constraint metrics measuring user interactions, the difference between Res-D2Q and VR is not statistically significant and thus can be safely ignored.

6 CONCLUSION

The surge in video consumption has been revolutionizing social media worldwide, causing increasing demand for optimizing the recommender systems on these video platforms. It remains to be a key problem to accurately predict the watch-time when evaluating different candidate videos, such that videos with potentially larger watch time are recommended to improve user engagement. However, watch-time prediction has been plagued by duration bias that is overlooked in existing models. Such bias originates from the goal of recommender systems to improve user watch time, such that spurious correlation between duration and watch time may be over-utilized and videos with long duration would be unfairly favored despite that they may be less aligned with user interests. Indeed, many video-sharing platforms are progressively biased towards videos with long duration. This issue becomes more severe due to the feedback loop of recommender systems, resulting in bias amplification that harms personalization.

In this paper, we formulate the problem of watch-time prediction using a causal graph, which characterizes the confounding effects of duration on both video exposure and watch-time prediction. We propose a *D*uration-*D*econfounded *Q*uantile-based (D2Q) framework, such that the natural effect of duration on watch-time is preserved, and the bias of duration on video is removed. Through extensive offline evaluation and live experiments, we demonstrate the advantages of our approach over alternatives in providing more accurate watch-time estimations, which further improves video consumption in real-time recommendation on Kuaishou App. We also vary the number of duration groups and show that our model performance improves first and then declines, due to deconfounding duration and reduced sample size respectively.

REFERENCES

[1] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2212–2220.

[2] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval.* 405–414.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems.* 104–112.

[4] 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).

[5] Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. 2021. End-to-End User Behavior Retrieval in Click-Through RatePrediction Model. *arXiv preprint arXiv:2108.04468* (2021).

[6] Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H Chi, and Minmin Chen. 2020. Deconfounding User Satisfaction Estimation from Response Rate Bias. In *Fourteenth ACM Conference on Recommender Systems.* 450–455.

[7] 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.

[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 *Proceedings of the fourth ACM conference on Recommender systems.* 293–296.

[9] Kuaishou financial reports 2021. 2022. Taylor series. https://newsfile.futunn.com/notice/2021/09/17/9941452-0.PDF. Online; accessed 08-Jan-2022.

[10] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards long-term fairness in recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining.* 445–453.

[11] 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 *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* 420–428.

[12] Daya Guo, Jiangshui Hong, Binli Luo, Qirui Yan, and Zhangming Niu. 2019. Multi-modal representation learning for short video understanding and recommendation. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW).* IEEE, 687–690.

[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] Shantanu Gupta, Zachary C Lipton, and David Childers. 2021. Estimating treatment effects with observed confounders and mediators. In *Uncertainty in Artificial Intelligence.* PMLR, 982–991.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[16] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.

[17] Thuc Duy Le, Lin Liu, Anna Tsykin, Gregory J Goodall, Bing Liu, Bing-Yu Sun, and Jiuyong Li. 2013. Inferring microRNA–mRNA causal regulatory relationships from expression data. *Bioinformatics* 29, 6 (2013), 765–771.

[18] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th acm international conference on information and knowledge management.* 2243–2251.

[19] Ruth M Mickey and Sander Greenland. 1989. The impact of confounder selection criteria on effect estimation. *American journal of epidemiology* 129, 1 (1989), 125–137.

[20] Judea Pearl. 2009. *Causality.* Cambridge university press.

[21] Judea Pearl. 2012. The do-calculus revisited. *arXiv preprint arXiv:1210.4852* (2012).

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

[23] Miquel Porta. 2008. *A dictionary of epidemiology.* Oxford university press.

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

[25] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web.* 521–530.

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

[27] Harald Steck. 2018. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems.* 154–162.

[28] Linpeng Tang, Qi Huang, Amit Puntambekar, Ymir Vigfusson, Wyatt Lloyd, and Kai Li. 2017. Popularity prediction of facebook videos for higher quality streaming. In *2017 {USENIX} Annual Technical Conference ({USENIX} {ATC} 17).* 111–123.

[29] Peng Wang, Yunsheng Jiang, Chunxu Xu, and Xiaohui Xie. 2019. Overview of Content-Based Click-Through Rate Prediction Challenge for Video Recommendation. In *Proceedings of the 27th ACM International Conference on Multimedia.* 2593–2596.

[30] Qi Wang, Dongmei Hao, Fangbai Li, Xiaoying Guan, and Pengcheng Chen. 2020. Development of a new framework to identify pathways from socioeconomic development to environmental pollution. *Journal of Cleaner Production* 253 (2020), 119962.

[31] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 1717–1725.

[32] 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.

[33] Neil A Weiss, Paul T Holmes, and Michael Hardy. 2006. *A course in probability.* Pearson Addison Wesley Boston, Massachusetts, USA.

[34] Wikipedia. 2022. Taylor series. https://en.wikipedia.org/wiki/Taylor_series. Online; accessed 08-Jan-2022.

[35] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. 2021. Fairrec: fairness-aware news recommendation with decomposed adversarial learning. AAAI.

[36] Siqi Wu, Marian-Andrei Rizoiu, and Lexing Xie. 2018. Beyond views: Measuring and predicting engagement in online videos. In *Twelfth international AAAI conference on web and social media.*

[37] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. Top-N Recommendation with Counterfactual User Preference Simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 2342–2351.

[38] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020.* 2849–2855.

[39] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. *arXiv preprint arXiv:2105.06067* (2021).

[40] 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 *Proceedings of the Web Conference 2021.* 2980–2991.

[41] 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.

A WEIGHTED LOGISTIC REGRESSION FOR WATCH-TIME PREDICTION

A.1 YouTube Method

Watch-time prediction via weighted logistic regression is first proposed by YouTube [7], where the expected watch time is calculated via the intermediate metric *Odds*. For completeness, we review their method here. Odds is defined by Equation 6, where k is the number of "positive" (which shall be defined shortly) samples and n is total number of samples. T_i is the watch time for "positive" video i. In YouTube recommender system, "positive" videos are those that are clicked by users and thus impressed, and "negtive" indicates the recommended video is not clicked. p_w is the predicted watch time.

$$\text{Odds} = \frac{\sum_i T_i}{N - k} = \frac{p_w}{1 - p_w}. \tag{6}$$

The calculation of *Odds* can be connected with that of expected watch time $\mathbb{E}(T)$ using Equation 7. p_{ctr} is the click-through-rate of the YouTube recommender system.

$$\begin{aligned} \text{Odds} &= \frac{\sum_i T_i}{N - k} = \frac{\sum_i T_i/N}{(N-k)/N} = \frac{\mathbb{E}(T)}{1 - k/N} \\ &= \frac{\mathbb{E}(T)}{1 - p_{\text{ctr}}}. \end{aligned} \tag{7}$$

The Taylor expansion [34] of $\frac{1}{1-x}$ is $1 + x + x^2 + x^3 + \cdots$. Thus Equation 7 can be rewritten as Equation 8, where the approximation is satisfied because p_{ctr} is around 0.01 such that p_{ctr}^2 is small enough.

$$\begin{aligned} \text{Odds} &= \frac{\mathbb{E}(T)}{1 - p_{\text{ctr}}} \\ &= \mathbb{E}(T) * (1 + p_{\text{ctr}} + p_{\text{ctr}}^2 + p_{\text{ctr}}^3 + \cdots) \\ &\approx \mathbb{E}(T) * (1 + p_{\text{ctr}}) \\ &\approx \mathbb{E}(T). \end{aligned} \tag{8}$$

Combining Equation 8 and Equation 6, the expected watch time can be calculated with Equation 9.

$$\mathbb{E}(T) \approx \frac{p_w}{1 - p_w}. \tag{9}$$

A.2 WLR: Adapting YouTube Method to Our Scenario

In contrast to the YouTube scenario where users have "click" action and then "watch" [7], short video platforms such as Kuaishou App serve users in a "top-down" scenario where they present videos in a full-screen and single-column format; thus the click-through-rate p_{ctr} in Equation 7 is not well-defined. Instead, we determine the "positive" and "negtive" labels based on whether the watch time surpasses the q_{60}-quantile of the empirical watch-time distribution. Following [7], positive samples are weighted by watch-time and negative samples receive unit weight. We can get similar result as Equation 7, which is shown in Equation 11.

$$\begin{aligned} \text{Odds} &= \frac{\sum_i T_i}{N - k} = \frac{\sum_i T_i/N}{(N-k)/N} = \frac{\mathbb{E}(T)}{1 - k/N} \\ &= \frac{\mathbb{E}(T)}{1 - p_{\geq q_{60}}}. \end{aligned} \tag{10}$$

Unfortunately, the average $p_{\geq q_{60}}$ is 0.4 and can not be approximated as p_{ctr} like Equation 8. Thus the final expected watch time in "top-down" scenario can be obtained by Equation 11.

$$\mathbb{E}(T) = \frac{p_w}{1 - p_w}(1 - p_{\geq q_{60}}), \tag{11}$$

where $p_{\geq q_{60}}$ is the predicted probability of the watch time surpassing the q_{60} quantile and can be obtained via minimizing the classical log-loss (see Equation 12) for binary classification.

$$\begin{aligned} \mathcal{L}(\theta) = -\frac{1}{N} \sum_{j=1}^{N} \Big(&y_{\geq q_{60}}(j) * log(p_{\geq q_{60}}(j)) \\ &+ (1 - y_{\geq q_{60}}(j)) * log(1 - p_{\geq q_{60}}(j)) \Big). \end{aligned} \tag{12}$$

B NETWORK ARCHITECTURE IN EXPERIMENTS

Recall that all methods in our experiment share the same network structure except for Res-D2Q that additionally has a MLP for duration adjustment. The overall network architecture can be divided into three parts: input layer, encoding layer, and output layer.

B.1 Input Layer

The goal of input layer is to embed different features, which can be divided into two categories: dense input x_{dense} and id input x_{id}. Particularly, we use $x_{duration}$ to denote the raw value of video duration for its importance in our work. The output of the embedding layer is $E_{dense} \in \mathbb{R}^{B \times 32}$, $E_{id} \in \mathbb{R}^{B \times 512}$ and $E_{duration} \in \mathbb{R}^{B \times 32}$, where B is the size of mini batch. In our experiments, we set $B = 512$.

B.2 Encoding Layer and Output Layer

The output of input layer is concatenated together, which is further transformed by a projection matrix $W \in \mathbb{R}^{576 \times 512}$ into an embedding matrix E:

$$E = \text{Concat}(E_{dense}, E_{id}, E_{duration})W. \tag{13}$$

Then the embedding matrix E is fed into three-layer of MLP whose embedding size is $\{256, 128, 64\}$ respectively, generating a hidden matrix $H \in \mathbb{R}^{B \times 64}$:

$$H = \text{Swish}\Big(\text{Swish}(EW^1 + b^1)W^2 + b^2\Big)W^3 + b^3. \tag{14}$$

Finally, H is fed into the output layer, which is output directly for VR and is activated with sigmoid function for others.

$$y = [y_w(1), y_w(2), \cdots]^\top = \text{Sigmoid}(H). \tag{15}$$

We note that the duration adjustment for Res-D2Q also happens in this layer, where H is additionally concatenated with the output from duration tower.