CREAD: A Classification-Restoration Framework with Error Adaptive Discretization for Watch Time Prediction in Video Recommender Systems

Jie Sun, Zhaoying Ding, Xiaoshuang Chen, Qi Chen, Yincheng Wang, Kaiqiao Zhan, Ben Wang*

Kuaishou Technology, Beijing, China

jie.sun327@gmail.com,{dingzhaoying, chenxiaoshuang, chenqi11, wangyincheng, zhankaiqiao, wangben}@kuaishou.com

Abstract

The watch time is a significant indicator of user satisfaction in video recommender systems. However, the prediction of watch time as a target variable is often hindered by its highly imbalanced distribution with a scarcity of observations for larger target values and over-populated samples for small values. State-of-the-art watch time prediction models discretize the continuous watch time into a set of buckets in order to consider the distribution of watch time. However, it is highly uninvestigated how these discrete buckets should be created from the continuous watch time distribution, and existing discretization approaches suffer from either a large learning error or a large restoration error. To address this challenge, we propose a Classification-Restoration framework with Error-Adaptive-Discretization (CREAD) to accurately predict the watch time. The proposed framework contains a discretization module, a classification module, and a restoration module. It predicts the watch time through multiple classification problems. The discretization process is a key contribution of the CREAD framework. We theoretically analyze the impacts of the discretization on the learning error and the restoration error, and then propose the error-adaptive discretization (EAD) technique to better balance the two errors, which achieves better performance over traditional discretization approaches. We conduct detailed offline evaluations on a public dataset and an industrial dataset, both showing performance gains through the proposed approach. Moreover, We have fully launched our framework to Kwai App, an online video platform, which resulted in a significant increase in users' video watch time by 0.29% through A/B testing. These results highlight the effectiveness of the CREAD framework in watch time prediction in video recommender systems.

1 Introduction

Recommender systems have seen great success in matching users with items they are interested in (Herlocker et al. 2004). One of the most popular applications is short-video social media (Tang et al. 2017; Wu, Rizoiu, and Xie 2018) such as TikTok and Instagram Reels, where short videos appear on the user's screens without any active operation such as clicking. Consequently, traditional metrics such as click-through rates are non-applicable anymore. Intuitively, watch time becomes a crucial metric for measuring user engagement (Covington, Adams, and Sargin 2016). To ensure optimal user experience, it is essential to predict watch time in online recommender systems accurately. By doing so, these platforms can gain a better understanding of user preferences and make personalized video recommendations tailored to their interests. A large body of research (Zhan et al. 2022; Gong et al. 2022; Lin et al. 2022; Wang et al. 2022; Cai et al. 2023; Zhao et al. 2023) has been devoted to developing neural network models, and significantly improves the prediction accuracy over traditional regression methods.

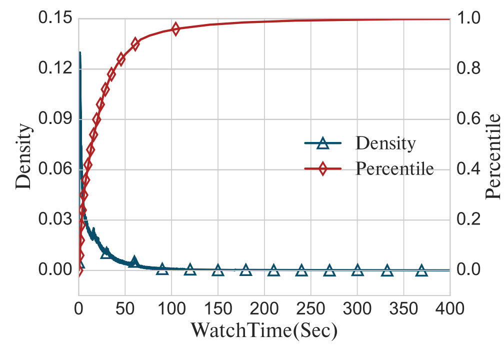

Figure 1: Probability density plot of watch time.

The prediction of watch time poses a regression problem due to the continuous and wide-ranging nature of users' watch time values, which amplifies susceptibility to outliers and potential prediction skew. The distribution of watch time for short videos, as depicted in Figure 1, is right-skewed, featuring substantial concentrations within limited time frames: 30% of views within 3 seconds and 80% within 32 seconds. This distribution presents a challenge for earlier watch time prediction attempts (Zhan et al. 2022; Covington, Adams, and Sargin 2016), which neglected the long-tailed nature of regression problems, thus yielding suboptimal results for tail instances. Moreover, in recommendation systems, ordinal relationships among predictions play a key role, with watch time as a metric for video comparison, highlighting the importance of ordinal ranking. However, standard regression losses like ℓ_1 and ℓ_2 do not consider ranking comparisons, focusing solely on discrepancy magnitude. Consequently, maintaining prediction accuracy and ranking

*Corresponding Author

efficacy within real-time recommender systems amid imbalanced continuous label distributions poses great challenges.

To address this, we introduce an effective classification-restoration-based framework for learning from imbalanced continuous targets in real-world contexts. The framework contains three key components: an effective discretization to convert continuous labels into ordinal intervals, a classification module training multiple binary classifiers across these segments to ensure ranking accuracy and a restoration module for predicting the watch time from the predictions of these classifiers.

A challenge with this approach is the ambiguity in creating discrete classes from the continuous distribution. This entails addressing two error types: the *learning error*, tied to sample bucket count, and the *restoration error*, which impacts watch time estimation from discretized predictions. Balancing these errors proves complex; narrow bucket intervals lower the learning error by reducing sample probability, while wider intervals diminish information and elevate the restoration error. We examine discretization's effect on learning and restoration errors and propose an error-adaptive discretization (EAD) method that harmonizes these errors within practical distributions.

Our comprehensive framework, named Classification-Restoration with EAD (CREAD), offers a versatile extension applicable to existing learning methods, such as D2Q. In summary, our main contributions are:

- We propose a general classification-restoration framework for learning the ordinal information of watch time, together with the training algorithm to reduce the classification and restoration error.
- We analyze the error bounds introduced by discretization, and propose a novel discretization approach to balance between the learning error and the restoration error, according to the realistic dataset distribution.
- Both offline and online experiments with realistic large-scale datasets show that our framework achieves competitive results compared with state-of-the-art methods.

2 Related Works

2.1 Watch Time Prediction

Watch time prediction is the task of predicting the watch time of a user, given the user profile, interactive history, and a series of candidate short videos. Value Regression (VR) directly predicts the absolute value of watch time, where the learned function's accuracy is evaluated by mean squared error. Covington, Adams, and Sargin (2016) incorporates watch time as a sample weight in the logistic regression of (WLR) impressive videos, transforming the direct regression of watch time into learned odds of video click-through rate. However, this assumption holds only when the impression rate is low and is not applicable in short video settings where impressed videos are played automatically. Recently, Zhan et al. (2022) studied duration bias in watch time prediction for video recommendation (D2Q), removing undesired bias by binning data based on duration. Although their equal-frequency-based methods removed the bias, they ignored the impact of the imbalanced distribution of watch time on long-tailed samples, resulting in lower accuracy compared to head samples. Although effective, they did not leverage the additional bucket-wise information lost in the discretization process, while we impose an error-adaptive framework in the modeling process.

2.2 Regression by Classification

Recently, there is a trend to pose the regression problem as a set of classification problems, which has a significant improvement on direct regression. The first related work is Ordinal Regression (OR). It is initially used for classification problems where the dependent variable exhibits a relative ordering and later extended to multiple fields, including age estimation (Niu et al. 2016; Beckham and Pal 2017) and depth estimation (Diaz and Marathe 2019). A vital problem of OR is the ambiguity in how the discrete classes should be created from the distribution. Most works use fixed standard methods such as equal-width or equal-frequency discretization to divide the continuous variables, while others manually choose multiple thresholds (Crammer and Singer 2001; Shashua and Levin 2002) as hyperparameters. As will be analyzed in Sec. 4, these methods introduce a large discretization error, especially when the data follows an imbalanced distribution. In contrast, our approach alleviates the problem by proposing an adaptive discretization method that minimizes the total error.

3 Method

Notations. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ be the training set, where $y_i \in \mathcal{Y} \subset \mathbb{R}^+$ is the corresponding ground truth watch time, and $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the i-th input, including user-related features (such as demographics characteristics and browsing histories) and video-related features (such as the tags and forwarding counts). Without loss of generality, we assume the domain of target variables is bounded by $T_{\max} \in \mathbb{R}^+$. We use $M-1$ thresholds $\{t_m\}_{m=1}^{M-1}$ to divide the domain into M discrete buckets $\mathcal{D} \triangleq \{d_m\}_{m=1}^M$, where the m-th bucket $d_m = [t_{m-1}, t_m)$ for $m = 1, \cdots, M$, and $t_0 = 0$, $t_M = T_{\max}$. Let $\hat{y}_i$ denote the predicted watch time of $\boldsymbol{x}_i$. Let $\mathbf{1}(\cdot)$ denote the indicator function. For simplicity, we omit the subscript i when we do not specify a certain sample.

3.1 Overall Framework

The CREAD Framework, as shown in Figure 2, includes three modules, *i.e.,* discretization, classification, and restoration. In the following, we explain the design of each component.

Discretization This module is a preprocess module independent of the training and evaluation process. It obtains thresholds $\{t_m\}_{m=1}^{M-1}$ according to the data distribution and break the target domain $\mathcal{Y}$ into M non-overlapping buckets $\mathcal{D} \triangleq \{d_m = [t_{m-1}, t_m)\}_{m=1}^M$. These buckets are used to transform the watch time y into m discretized labels:

$$y_m = \mathbf{1}(y > t_m). \tag{1}$$

The discretization strategy is crucial for prediction accuracy, and we will discuss it in detail in Sec. 4.3.

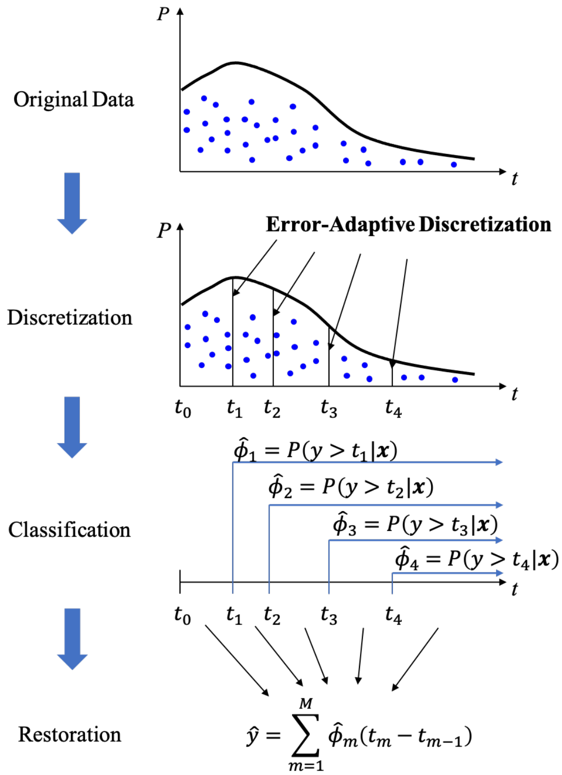

Figure 2: The CREAD Framework.

Classification M classifiers are trained to predict whether the watch time y is larger than the m-th threshold t_m, *i.e.*, y_m in Eq. (1), and output a series of probabilities:

$$\hat{\phi}_m(\boldsymbol{x}_i; \Theta_m) = P(y > t_m|\boldsymbol{x}_i),\ 1 \le i \le N. \tag{2}$$

The classifiers are neural networks with learnable parameters Θ_m. We introduce how to train the models in Sec. 3.2.

Restoration Given the $\{\hat{\phi}_m\}_{m=1}^M$, we are able to restore the predicted watch time. The restoration is based on the following fact of expectation:

$$\begin{aligned} \mathbb{E}(y|\boldsymbol{x}_i) &= \int_{t=0}^{t_M} tP(y = t|\boldsymbol{x}_i)dt \\ &= \int_{t=0}^{t_M} P(y > t|\boldsymbol{x}_i)dt \\ &\approx \sum_{m=1}^{M} P(y > t_m|\boldsymbol{x}_i)\,(t_m - t_{m-1})\,. \end{aligned} \tag{3}$$

By the definition of $\hat{\phi}_m$ in Eq. (2), we can reconstruct the predicted watch time from these discretized predictions $\hat{\phi}_m$:

$$\hat{y} = \sum_{m=1}^{M} \hat{\phi}_m\,(t_m - t_{m-1})\,. \tag{4}$$

3.2 Model Training

Here, we provide the loss function in the training of M classifiers. The loss functions contain three parts, of which the first is the **classification loss** via standard cross-entropy.

$$\mathcal{L}_{\text{ce}} = \sum_{m=1}^{M} -y_m \log(\hat{\phi}_m) - (1 - y_m)\log(1 - \hat{\phi}_m). \tag{5}$$

The second is the **restoration loss** to reduce the error of the reconstructed watch time in Eq. (4):

$$\mathcal{L}_{\text{restore}} = \ell(\hat{y}, y), \tag{6}$$

where ℓ is a loss function to measure the deviation from $\hat{y}$ to y. We find it beneficial to apply Huber loss(Huber 1992) for $\mathcal{L}_{\text{restore}}$, which will be detailed analyzed in Sec. 5.3.

The third is a **regularization term by ordinal prior**. By definition, the output of the M classifiers $\{\phi_m\}_{m=1}^M$ has a prior by definition, *i.e.*, $\hat{\phi}_m$ monotonically decreases as m grows. Consequently, we impose the prior into the proposed framework by minimizing the following regularization:

$$\mathcal{L}_{\text{ord}} = \sum_{m=1}^{M-1} \max(\hat{\phi}_{m+1} - \hat{\phi}_m, 0). \tag{7}$$

In summary, the final optimization objective is

$$\mathcal{L}_{\text{CREAD}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{restore}}\mathcal{L}_{\text{restore}} + \lambda_{\text{ord}}\mathcal{L}_{\text{ord}}, \tag{8}$$

where λ_{ce}, λ_{restore}, and λ_{ord} are hyperparameters.

3.3 Challenges of Discretization

In the CREAD framework, a key module is the discretization module, and the method of discretization largely affects the final prediction accuracy. As shown in Figure 3, the discretization introduces two kinds of errors:

- **Learning Error**: Since the number of instances in each bucket is finite, the M classifiers cannot be infinitely accurate. As we increase the number of buckets M, the number of instances that fall in each bucket decreases, thus limiting the classification performance.
- **Restoration Error**: The restoration in Eq. (4) is an approximate function of the expectation, omitting the detailed probability density in each bucket $[t_{m-1}, t_m]$, which will also introduce errors.

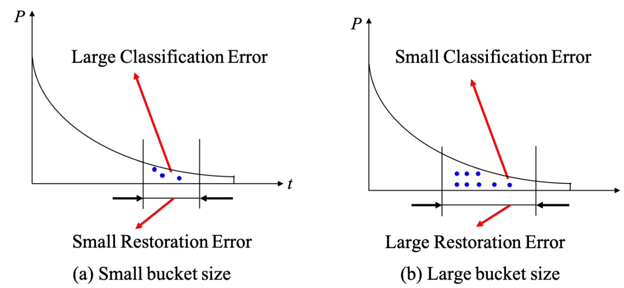

Figure 3: Two kinds of errors in the discretization.

Unfortunately, these two errors cannot be simultaneously reduced. To reduce the learning error, a larger bucket width is needed, leading to a larger restoration error (see Figure 3). Existing approaches usually use equal-width or equal-frequency methods (Gai et al. 2017) to heuristically set the discretization set $\mathcal{D}$. We will show in Sec. 4 that both equal-width and equal-frequency methods cannot balance the two errors well, and propose our EAD approach.

4 Balance the Errors in Discretization

This section aims to balance the errors introduced in the discretization process. We first provide a theoretical analysis of the error bounds of the learning error and the restoration error introduced in the discretization process, and then propose the EAD approach to effectively balance the two errors.

4.1 Decomposition of Errors from Discretization

Assume the training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \sim \mu(\boldsymbol{x}, y) = \mu(\boldsymbol{x})\mu(y|\boldsymbol{x})$ is *i.i.d.* distributed. Let $p_m(\boldsymbol{x}) = P(y \in d_m|\boldsymbol{x})$ denote the probability that the label y belongs to the m-th bucket d_m given $\boldsymbol{x}$. Let $v_m(\boldsymbol{x}) = \mathbb{E}(y|\boldsymbol{x}, y \in d_m)$ be the expectation of the watch time of sample $\boldsymbol{x}$, given that it belongs to the m-th bucket. Let $w_m = \mathbb{E}_{\boldsymbol{x}\sim\mu(\mathbf{x})} v_m(\boldsymbol{x})$ denote the expectation of watch time in the interval d_m.

We add the hat superscript to denote the predicted value, *e.g.,* $\hat{p}_m(\boldsymbol{x})$ as the prediction of $p_m(\boldsymbol{x})$ and $\hat{w}_m$ as the prediction of w_m. Then we can formulate the watch time as $\hat{y} = \sum_m \hat{p}_m(\boldsymbol{x})\hat{w}_m$. Notice that such form is equivalent to the cumulative form in Eq. (4), where $\hat{p}_m = \hat{\phi}_m - \hat{\phi}_{m-1}$.

Now we aim to estimate the error bound between the predicted watch time $\hat{y}$ and the ground truth y. To achieve this, we first provide a decomposition of the errors:

Lemma 4.1. *Assume that $\hat{p}_m(\boldsymbol{x})$ and $\hat{w}_m$ are unbiased estimations of $p_m(\boldsymbol{x})$ and w_m, respectively, we have*

$$\mathbb{E}\left(\hat{y} - y\right)^2 = V_p + V_w + V_b + V_y, \tag{9}$$

where

$$V_p = \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\hat{p}}\mathbb{E}_{\hat{w}}\left[\sum_m \left(\hat{p}_m(\boldsymbol{x}) - p_m(\boldsymbol{x})\right)\hat{w}_m\right]^2, \tag{10}$$

$$V_w = \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\hat{w}}\left[\sum_m p_m(\boldsymbol{x})\left(\hat{w}_m - w_m\right)\right]^2, \tag{11}$$

$$V_b = \mathbb{E}_{\boldsymbol{x}}\left[\sum_m p_m(\boldsymbol{x})\left(w_m - v_m(\boldsymbol{x})\right)\right]^2, \tag{12}$$

$$V_y = \mathbb{E}_{\boldsymbol{x},y}\left[\sum_m p_m(\boldsymbol{x})v_m(\boldsymbol{x}) - y\right]^2. \tag{13}$$

Please refer to Appendix. A for the detailed proof. Intuitively, V_p is determined by the learning error of the probability that y falls into each bucket, i.e., $p_m(\boldsymbol{x})$. V_w describes the impacts of learning error on the representing value $\hat{w}_m$, V_b is the error caused by reconstructing the watch time from discretization, and V_y is the intrinsic variance of the watch time y.

The two prediction errors V_p and V_w are affected by the error of the learning algorithm. Therefore, these two error terms correspond to the learning error. In contrast, V_b corresponds to the restoration error independent of the concrete learning algorithms. Finally, V_y is unrelated to either the learning or the discretization processes, and will not be discussed later.

4.2 Error Bounds of Discretization

This section analyzes how the discretization process affects the error bounds. For ease of theoretical analysis, we only consider the case of tabular input and assume that $\mu(\boldsymbol{x}, y)$ is sufficiently smooth. But we emphasize that the discretization approach inspired by the theoretical analysis will also be effective in real-world settings.

Theorem 4.2. *Assume that the input $\boldsymbol{x}$ is sampled from a finite set $\mathcal{X}$. Moreover, assume $\hat{p}_m(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$ and $\hat{w}_m$ are obtained from maximum likelihood estimation. Moreover, assume $\mu(\boldsymbol{x}, y)$ is with a bounded second partial derivative. Then we have*

$$V_p \le \overline{V}_p \triangleq \frac{C_p\,|\mathcal{X}|}{N} \cdot A_p(\mathcal{D}), \tag{14}$$

$$V_w \le \overline{V}_w \triangleq \frac{C_w}{N} \cdot A_w(\mathcal{D}), \tag{15}$$

$$V_b \le \overline{V}_b \triangleq C_b \cdot A_b(\mathcal{D}), \tag{16}$$

where C_p, C_w and C_b are constants independent of the discretization $\mathcal{D}$, and A_p, A_w, A_p are functions of $\mathcal{D}$:

$$A_p(\mathcal{D}) = M\mathbb{E}_{y\sim\Psi} y^2, \tag{17}$$

$$\begin{aligned} A_w(\mathcal{D}) = &\sum_{m\in\mathcal{M}} \left[\Psi(t_m) - \Psi(t_{m-1})\right]^2 \\ &\cdot \sum_{m\in\mathcal{M}} \frac{\left(t_m - t_{m-1}\right)^2}{\Psi(t_m) - \Psi(t_{m-1})}, \end{aligned} \tag{18}$$

$$\begin{aligned} A_b(\mathcal{D}) = &\sum_{m\in\mathcal{M}} \left[\Psi(t_m) - \Psi(t_{m-1})\right]^2 \\ &\cdot \sum_{m\in\mathcal{M}} \left(t_m - t_{m-1}\right)^2, \end{aligned} \tag{19}$$

where Ψ is the CDF of watch time y:

$$\Psi\left(t\right) \triangleq P\left\{y \le t\right\} = \mathbb{E}_{\boldsymbol{x}} \int_0^t \mu(y|\boldsymbol{x})dy. \tag{20}$$

Proof. See Appendix B. □

Therefore, we find that the prediction error is bounded by several functions A_p, A_w and A_b, which only depend on the watch time distribution Ψ and the discretization $\mathcal{D}$. Now we provide some intuitive explanations.

Discussion What are the impacts of different discretization methods on each error term? Here we mainly discuss the error terms A_w and A_b, since they depend on the division points $\{t_m\}_{m=1}^M$ of $\mathcal{D}$. We consider a truncated exponential distribution on $[0, 1]$, *i.e.,* $\Psi(t) = (1 - e^{-5t})/(1 - e^{-5})$, discretized by 10 buckets. Table 1 shows the different terms on the equal-width and equal-frequency discretization. It is shown that the equal-width method leads to lower A_b, while the equal-frequency method leads to lower A_w. This result has a very intuitive explanation, which shows the meaning of A_b and A_w:

- **Learning Error**: The learning error is shown by A_w, which is affected by the number of samples in each bucket. Specifically, there exists a $\Psi(t_m)-\Psi(t_{m-1})$ term in the denominator of A_w. Therefore, if there are few samples in some buckets, the corresponding probability $\Psi(t_m) - \Psi(t_{m-1})$ will be small, leading to a large error.

Method	$A_w(\mathcal{D})$	$A_b(\mathcal{D})$
Equal-Width	1.42	**0.025**
Equal-Freq	**0.34**	0.034

Table 1: Different Error Terms on Different Methods.

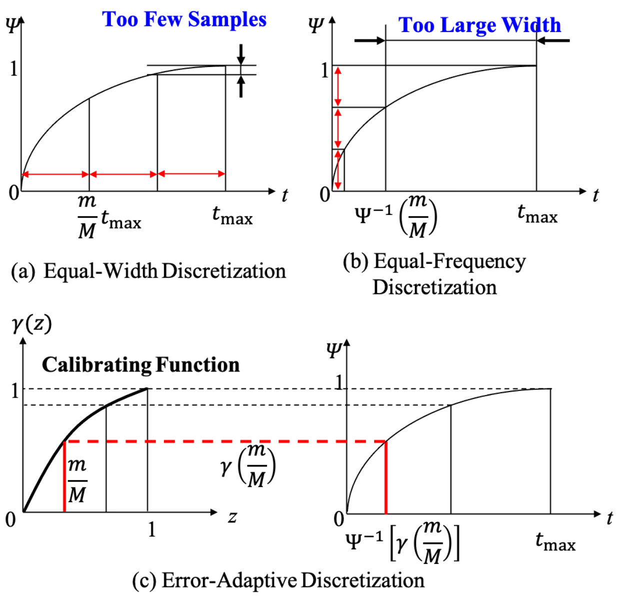

Figure 4: EAD compared with traditional methods.

- **Restoration Error**: A_b shows the restoration error bound. It contains a $t_m - t_{m-1}$ term as a multiplier. When $t_m - t_{m-1}$ is large for some m, the error term will increase, which is consistent with the intuition that larger bucket width will lead to a larger restoration error.

Now we discuss again the dilemma of errors introduced by discretization: **the learning error and the restoration error usually contradict**. If we are about to decrease the learning error, we need to increase the samples in each bucket, but a large bucket width leads to a large restoration error. Formally, the bucket probability $\Psi(t_m) - \Psi(t_{m-1})$ is usually positively related to the bucket width $t_m - t_{m-1}$. Given the abovementioned discussions, we are about to provide the EAD approach to balance the two kinds of errors.

4.3 EAD Approach

Here we mainly discuss the discretization strategy $\mathcal{D}$ given the number of buckets M. We do not discuss the choice of M since it is a single variable that can be regarded as a hyperparameter. According to Sec. 4.2, the discretization strategy $\mathcal{D}$ needs to balance the learning error A_w and the restoration error A_b. Thus, the discretization strategy of EAD minimizes the following objective:

$$\min_{\mathcal{D}} J(\mathcal{D}) = A_w(\mathcal{D}) + \beta A_b(\mathcal{D}), \tag{21}$$

where β is determined by C_w, C_b, and N according to Theorem 4.2. Since C_w and C_b depend on the characteristics of the datasets, which cannot be obtained theoretically, we regard β as a hyperparameter.

Eq. (21) is an optimization problem with dimension $M - 1$. It is challenging to find the optimal discretization strategy since M is usually tens or hundreds. Here we propose a more lightweight method. We begin by expressing the equal-width and equal-frequency discretization methods formally. Specifically, the equal-width method writes:

$$t_m = \frac{m}{M} T_{\max}, \tag{22}$$

which guarantees a fixed $\Delta t_m = T_{\max}/M$, but leads to too small $\Delta\Psi(t_m)$ in the long-tailed buckets (see Figure 4(a)). In contrast, the equal-frequency method writes:

$$t_m = \Psi^{-1}\left(\frac{m}{M}\right), \tag{23}$$

which guarantees a fixed $\Delta\Psi(t_m) = 1/M$, but leads to too large Δt_m in the long-tailed buckets (see Figure 4(b)).

The key is to rewrite Eq. (22)∼(23) by:

$$t_m = \Psi^{-1}\left[\gamma\left(\frac{m}{M}\right)\right], \tag{24}$$

where γ is a calibrating function $\gamma : [0,1] \to [0,1]$, with $\gamma(0) = 0, \gamma(1) = 1$. Note that when $\gamma(z) = \Psi(zT_{\max})$, we obtain Eq. (22); and when $\gamma(z) = z$, we obtain Eq. (23).

Therefore, Eq. (24) contains two extreme cases of the discretization methods. This inspires us that by properly choosing the calibrating function γ between the real distribution Ψ and the uniform distribution, we can obtain any intermediate bucketing strategies. Figure 4(c) shows the impacts of the calibrating function γ on the discretization process.

According to the abovementioned discussions, we set γ to be in a group of functions $\gamma(\cdot;\alpha)$, parameterized by α, then it is possible to find the optimal γ under the optimization problem in Eq. (21) by grid search.

As an illustration, we use the same settings as Discussion 2 in Sec. 4.2, and set γ to be $\gamma(z;\alpha) = (1 - e^{-\alpha z}) / (1 - e^{-\alpha})$ with a parameter α from 0 to 5, and set $\beta = 50/100/200$. Specifically, $\alpha = 0$ degrades to the equal-frequency method, while $\alpha = 5$ degrades to the equal-width method. Figure 5 shows the objective function $J(\mathcal{D})$ (Eq. (21)) under α and β, which shows

- The hyperparameter β allows us to balance the learning error and the restoration error flexibly.
- By comparing the optimal α with the equal-frequency method ($\alpha = 0$) and the equal-width method ($\alpha = 5$), it is shown that by properly choosing the calibrating function γ, it is possible to find a more proper bucketing over traditional equal-frequency and equal-width methods.

5 Experiments

In this section, we evaluate our CREAD framework on benchmark recommendation datasets, including two public recommendation datasets (CIKM[1] and KuaiRec (Gao et al. 2022)) and a realistic industrial dataset (Industrial). In addition, we perform ablation studies to validate the design of the proposed framework, the training strategy, and the selection of the hyperparameters. We refer to Appendix B for the details of the datasets, model architectures, baselines, more ablation studies, and training details.

[1]https://competitions.codalab.org/competitions/11161

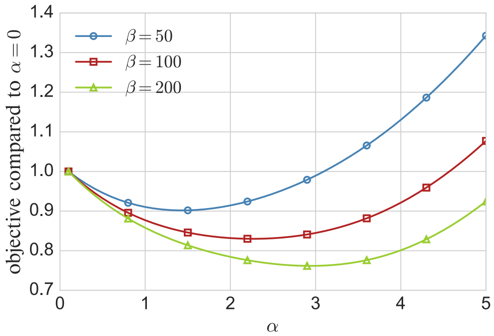

Figure 5: Error terms in different hyperparameters.

5.1 Experimental Settings

Datasets To make a fair comparison, we adopt the offline experiments on industrial datasets and two public datasets for reproducibility. The large-scale industrial dataset (referred to as *Industrial*) is collected from a real-world streaming short-video platform. The dataset encompasses roughly 6 billion impressions and over 50 million users each day. We accumulated users' interaction logs for 14 days, and subsequently use the next day's information for evaluation. Moreover, to improve reproducibility, we adopt public benchmarks dataset KuaiRec (Gao et al. 2022) and CIKM. The KuaiRec dataset is encompasses a total of $7,176$ users, $10,728$ distinct items, and a substantial count of $12,530,806$ impressions, while the evaluation dataset comprises $1,411$ users, $3,327$ items, and a collection of $4,676,570$ impressions. And the CIKM dataset contains $87,934$ users, $122,994$ items and $1,235,381$ impressions.

Evaluation Metrics We follow previous works (Zhan et al. 2022) and adopt the following metrics to measure watch time prediction quantitatively:

- **MAE (Mean Absolute Error)** is measured as the average of the absolute error between the watch time prediction $\hat{y}_i$ and the ground truth y_i, which is $\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$.
- **XAUC (Zhan et al. 2022)** is measured by randomly sampling instance pairs and checking whether their relative order is consistent with the ground truth.

Baselines Here, we compare our method with previous methods for the watch time prediction task. The reference methods can be classified as traditional machine learning approaches and neural network approaches. The former includes Value Regression (VR), Weighted Logistic Regression (WLR) (Covington, Adams, and Sargin 2016), and Ordinal Regression (OR), while the latter includes the state-of-the-art model D2Q (Zhan et al. 2022). In particular, VR and WLR are continuous regression approaches (denoted as Cont.), while OR and D2Q break the target range into discrete buckets (denoted as Disc.). Since the authors did not provide an official code for watch time prediction, we implement them according to their corresponding papers, tune the number of buckets (if applicable) and report the best results.

Data	Type	Cont.		Disc.		
	Method	VR	WLR	OR	D2Q	CREAD
Kuai.	XAUC	0.5333	0.5483	0.5883	0.5677	**0.6009**
	MAE	3.6992	3.5722	3.2946	3.3090	**3.2150**
CIKM	XAUC	0.6058	0.6310	0.6313	0.6568	**0.6671**
	MAE	2.3780	1.9785	1.9157	2.5499	**1.7928**
Indust.	XAUC	0.6324	0.6323	0.6344	0.6423	**0.6459**
	MAE	20.3672	20.2740	20.1263	19.0740	**18.6716**

Table 2: Watch time prediction results on KuaiRec, CIKM, and Industrial datasets: XAUC and MAE.

5.2 Comparison with State of the Art

In this section, we compare our method with current state-of-the-art methods for watch time prediction and report the MAE and XAUC metric in Table 2. We can see that the discrete approaches outperform the continuous regression approaches, indicating that the regression-by-classification framework is superior to direct regression in dealing with imbalanced data. Specifically, the discrete methods outperform continuous methods by at least 0.2632 in MAE and 0.0194 in XAUC on KuaiRec dataset. Besides, our CREAD outperforms all the discretization methods on these datasets with both metrics. Notably, CREAD outperforms the best (discretization) baseline D2Q by 5.85% on KuaiRec with XAUC, and 2.11% on Industrial with MAE.

5.3 Ablation Studies

We further conduct an ablation study on the proposed adaptive discretization, the ordinal regularization, and choices of the restoration function.

Effectiveness of Adaptive Discretization In Section 4.3, we explain the relation of our approach to the commonly used equal-width and equal-frequency discretization methods, as well as how our EAD balances them. To empirically verify this, we conduct an ablation study on KuaiRec by altering the discretization method and keeping other settings fixed. Specifically, we compare the equal-width discretization, equal-frequency discretization, and our adaptive discretization when $\beta \in \{0.5, 1.0, 3.0, 10.0\}$. As shown in Table 3, when we increase β from 0.5 to 10, the width of the tail bucket, *i.e.,* the maximal bucket width, decreases monotonically from 858.69s to 34.74s, always less than the width of the equal-frequency discretization, and more than that of the equal-width discretization. The same as the largest bucket width, the sample percentile of the tail bucket also reduces monotonically as β grows. Moreover, the XAUC and MAE metrics are consistent, and both have a peak performance with $\beta = 3.0$. Therefore, it is necessary to choose an appropriate β that adapts to the realistic distribution. We adopt $\beta = 3.0$ in all other experiments. Nevertheless, note that our EAD with any β (the worst performance is with $\beta = 0.5$) yields superior performance than equal-width discretization (-0.0073 in MAE) and equal-frequency discretization (-0.0058 in MAE). Finally, we observe that equal-frequency discretization achieves better performance (-0.0015 in MAE) than equal-width discretiza-

Metrics	*Equal Frequency*	$\beta = 0.5$	$\beta = 1.0$	$\beta = 3.0$	$\beta = 10.0$	*Equal Width*
width(s) of tail bucket	981.93	858.69	377.95	315.21	34.74	33.0
sample (%) of tail bucket	3.3%	0.1437%	0.0101%	0.0077%	0.0076%	0.0011%
XAUC (↑)	0.5962	0.5977	0.5992	**0.6009**	0.6001	0.5982
MAE (↓)	3.2477	3.2419	3.2451	**3.2150**	3.2335	3.2492

Table 3: Comparison on XAUC, MAE with different choices of label discretization on real-world data

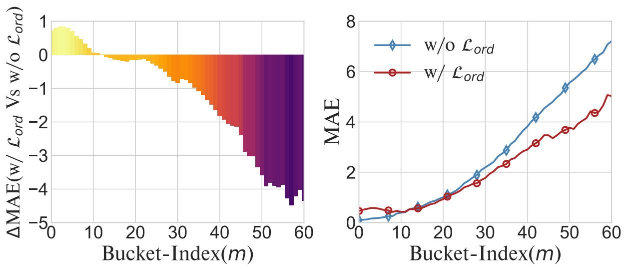

Figure 6: Comparison between loss w/o and w/ $\mathcal{L}_{ord}$ in MAE score with respect to bucket index.

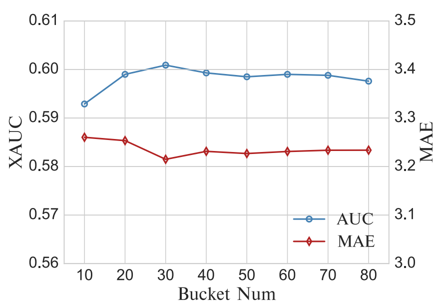

Figure 7: AUC and MAE of CREAD with different buckets.

tion when dealing with long-tailed data.

Effectiveness of Ordinal Regularization We verify the effectiveness of the ordinal regularization $\mathcal{L}_{\text{ord}}$ introduced in Sec. 3.2 by training a model without it on KuaiRec. Experimental results show that ordinal regularization effectively improves performance by 0.0132 in MAE, and 0.33 in XAUC. Moreover, we show the performance change relative to our CREAD (denoted as ΔMAE) as a function of the bucket index m and report the difference in Figure 6, from which we can draw several conclusions. First, whether with the ordinal regularization, MAE increases along the bucket index m, which is consistent with the long-tail distribution shown in Figure 1. Second, we observe that the ordinal regularization significantly reduces the error in the tail buckets. This is mainly because our EAD significantly reduces MAE in the tail buckets (Please refer to Appendix B.5), and our ordinal regularization further strengthens the effect. Finally, we attribute the performance degradation within the first 12 buckets as a sacrifice to the overall generality.

Days	Main Metric.	Constraint Metrics			
	Watch-Time	Like	Share	Follow	Comment
D1	+0.229%	+0.132%	-0.334%	+0.102%	+1.186%
D2	+0.275%	+0.230%	-0.334%	+0.364%	-1.143%
D3	+0.273%	+0.377%	-0.112%	+0.212%	+2.337%
D4	+0.327%	+0.002%	+0.221%	+0.310%	+1.781%
D5	+0.353%	-0.150%	+0.157%	+0.392%	-0.782%

Table 4: Online A/B results.

Influence of the Bucket Number We investigate the influence of the bucket number M in the discretization step. As shown in Figure 7, given the trade-off between the MAE and XAUC, we adopt $M = 30$ in all other experiments.

5.4 Online A/B Testing

Besides offline experiments, we also conduct an online A/B test on Kwai App, a streaming short-video platform with more than 50 million users each day. We attribute 20% traffic to our CREAD and the baseline D2Q (Zhan et al. 2022), respectively. We launch the experiments on the real-time system for 5 days and report the results in Table 4. As we can see, the treatment group experienced a noteworthy increase in watch time (referred to as *main metric*) by 0.291%. This outcome is particularly striking since production algorithms typically yield an average improvement of only 0.1% to 0.2%. Furthermore, we evaluate multiple metrics (referred to as *main metric*) commonly used in actual production systems, including the cumulative counts of *like* (clicking the like button), *follow* (subscribing to the video creator), *share* (disseminating the video among friends), and *comment* (providing textual remarks). The fluctuation of these metrics is not statistically significant.

6 Conclusion

In order to predict user watch time in large-scale recommender systems, we propose a novel discretization framework, called CREAD, to learn from the skewed distribution and ordinal information of watch time. We analyze the error bounds introduced by discretization and propose a novel discretization approach to balance the learning error and the restoration error. Moreover, we also propose an ordinal regularization to capture the sequential prior between the discrete buckets. For offline and online experiments, we demonstrate that our model significantly improves the recommendation performance.

References

Beckham, C.; and Pal, C. 2017. Unimodal probability distributions for deep ordinal classification. In *ICML*, 411–419.

Cai, Q.; Liu, S.; Wang, X.; Zuo, T.; Xie, W.; Yang, B.; Zheng, D.; Jiang, P.; and Gai, K. 2023. Reinforcing user retention in a billion scale short video recommender system. In *WWW*, 421–426.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep neural networks for youtube recommendations. In *RecSys*, 191–198.

Crammer, K.; and Singer, Y. 2001. Pranking with ranking. *NeurIPS*, 14.

Diaz, R.; and Marathe, A. 2019. Soft labels for ordinal regression. In *CVPR*, 4738–4747.

Gai, K.; Zhu, X.; Li, H.; Liu, K.; and Wang, Z. 2017. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194*.

Gao, C.; Li, S.; Lei, W.; Chen, J.; Li, B.; Jiang, P.; He, X.; Mao, J.; and Chua, T.-S. 2022. KuaiRec: a fully-observed dataset and insights for evaluating recommender systems. In *CIKM*, 540–550.

Gong, X.; Feng, Q.; Zhang, Y.; Qin, J.; Ding, W.; Li, B.; Jiang, P.; and Gai, K. 2022. Real-time short video recommendation on mobile devices. In *CIKM*, 3103–3112.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1): 5–53.

Hildebrand, F. B. 1987. *Introduction to numerical analysis*. Courier Corporation.

Huber, P. J. 1992. Robust estimation of a location parameter. In *Breakthroughs in Statistics: Methodology and Distribution*, 492–518. Springer.

Kingma, D. P.; and Ba, J. 2014. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, Z.; Wang, H.; Mao, J.; Zhao, W. X.; Wang, C.; Jiang, P.; and Wen, J.-R. 2022. Feature-aware diversified re-ranking with disentangled representations for relevant recommendation. In *SIGKDD*, 3327–3335.

Niu, Z.; Zhou, M.; Wang, L.; Gao, X.; and Hua, G. 2016. Ordinal regression with multiple output cnn for age estimation. In *CVPR*, 4920–4928.

Shashua, A.; and Levin, A. 2002. Ranking with large margin principle: Two approaches. *NeurIPS*, 15.

Tang, L.; Huang, Q.; Puntambekar, A.; Vigfusson, Y.; Lloyd, W.; and Li, K. 2017. Popularity prediction of facebook videos for higher quality streaming. In *USENIX ATC*, 111–123.

Wang, J.; Ma, W.; Li, J.; Lu, H.; Zhang, M.; Li, B.; Liu, Y.; Jiang, P.; and Ma, S. 2022. Make fairness more fair: fair item utility estimation and exposure re-distribution. In *SIGKDD*, 1868–1877.

Wu, S.; Rizoiu, M.-A.; and Xie, L. 2018. Beyond views: measuring and predicting engagement in online videos. In *ICWSM*.

Zhan, R.; Pei, C.; Su, Q.; Wen, J.; Wang, X.; Mu, G.; Zheng, D.; Jiang, P.; and Gai, K. 2022. Deconfounding duration bias in watch-time prediction for video recommendation. In *SIGKDD*, 4472–4481.

Zhao, W.; Tang, D.; Chen, X.; Lv, D.; Ou, D.; Li, B.; Jiang, P.; and Gai, K. 2023. Disentangled causal embedding with contrastive learning for recommender system. In *WWW*, 406–410.