Leveraging Label Distributions as Anchors to Enhance Video Recommendation

Yulin Xu*
University of California, Irvine
CA, USA
yulinxu895@gmail.com

Chao Cui*
Tsinghua University
Beijing, China
chaocui01@gmail.com

Shisong Tang*
KuaiShou Inc.
Beijing, China
tangshisong@kuaishou.com

Fan Li
KuaiShou Inc.
Beijing, China
lifan@kuaishou.com

Bing Han
KuaiShou Inc.
Beijing, China
hanbing@kuaishou.com

Huafeng Cao
Kuaishou Inc.
Beijing, China
caohuafeng@kuaishou.com

Jiechao Gao†
Center for SDGC, Stanford University
Palo Alto, CA, USA
jiechao@stanford.edu

Hechang Chen
Jilin University
Changchun, China
chenhc@jlu.edu.cn

Abstract

In video recommendation systems, accurately predicting watch time is crucial for enhancing user engagement and retention. Traditional methods typically apply label transformations or mitigate duration bias to improve performance but overlook that erroneous instance representations are the primary cause of significant prediction errors. Moreover, these approaches predominantly rely on point perdition, limiting their robustness. To address these challenges, we propose LDA, a novel prediction paradigm that optimizes instance representations by explicitly leveraging label distributions as anchors within the model, enabling more accurate and robust predictions. Our analysis reveals that watch ratio across different duration groups exhibit distinct multi-peak distributions, reflecting the strong aggregation of user behavior. Based on this finding, we employ Vector Quantized Variational Auto-encoder (VQ-VAE) to convert the continuous watch ratio distribution into representative anchors that capture these multi-peak characteristics within each duration group. Subsequently, we project both instance representations and anchors into a common space and utilize Optimal Transport (OT) to generate pseudo-labels aligned with the anchor distribution, allowing instances to obtain structured coordinates within this space during training. Finally, we derive optimized instance representations for watch time prediction by aggregating anchor vectors through weighted integration. Extensive offline experiments on two datasets and large-scale online A/B testing on a short-video platform with over 300 million DAUs demonstrate the consistent superiority of LDA in watch time prediction.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Video recommendation, Watch time prediction, Optimal transport

ACM Reference Format:
Yulin Xu, Chao Cui, Shisong Tang, Fan Li, Bing Han, Huafeng Cao, Jiechao Gao, and Hechang Chen. 2025. Leveraging Label Distributions as Anchors to Enhance Video Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3711896.3737276

1 Introduction

Recently, video sharing platforms (e.g., TikTok and Douyin) have emerged as some of the most popular applications, wherein their recommendation systems play a pivotal role in aligning user preferences with appropriate content [15, 22, 30, 31, 37, 40]. On these platforms, videos are automatically displayed on users' screens without requiring active clicks, rendering traditional metrics such as click-through rate less applicable in this scenario. Instead, watch time has ascended as a critical metric for evaluating users' interest in video content [16, 19, 23, 36, 43]. It not only provides an intuitive reflection of content quality and user experience but also plays a significant role in driving advertising strategies and personalized services. Consequently, accurately predicting watch time has emerged as a critical task in video recommendation systems[9, 31, 44].

In early research, YouTube introduced the Weighted Logistic Regression (WLR) method [3], which reformulated the watch time prediction problem into a weighted binary classification task, thereby providing an initial solution for the field. Subsequently, Zhan et.al proposed a D2Q [43] method tp achieved more precise predictions by transforming watch time. Lin et al. proposed the Tree-based Progressive Model (TPM) [20] method decomposed prediction into

*Equal contributions.
†Corresponding Author.

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737276

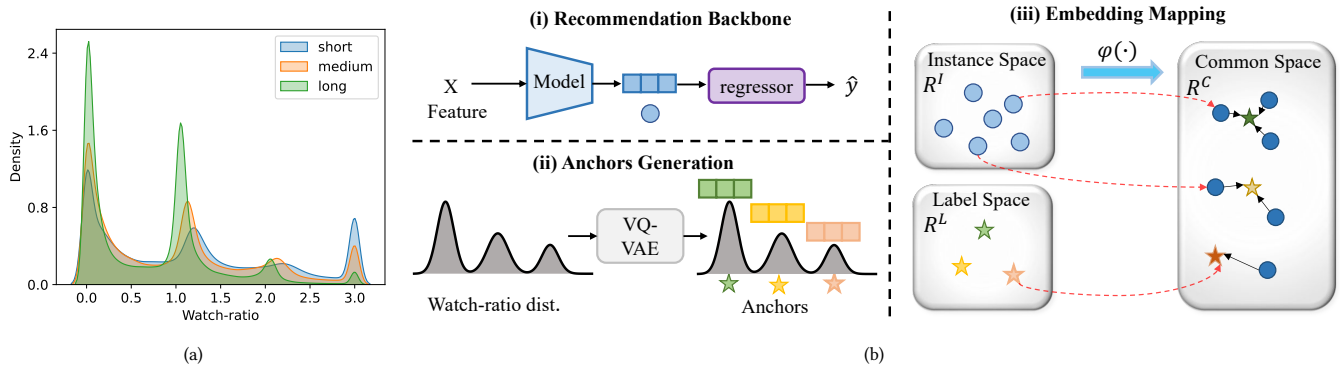


Figure 1: (a) The watch ratio distributions of different duration buckets exhibit distinct multi-peak patterns. (b) The core idea of our proposed LDA method.

a series of ordinal classification tasks with conditional dependencies using a tree structure, in order to capture the inherent ordinal relationships. Another line of research focuses on eliminating duration bias by employing causal inference frameworks to improve prediction performance; for example, the DVR, D^{2}Co, and CWM [46, 47, 49] methods adopt this technique, demonstrating its effectiveness.

However, these methods primarily focus on label transformation or eliminating duration bias, overlooking the fact that erroneous instance representations within the model are the fundamental cause of prediction errors. Moreover, relying on point predictions prevents capturing the overall structure of user behavior in complex label distributions, undermining robustness.

To address the aforementioned issues, we first analyzed, from a mathematical perspective, the relationship between prediction error and the quality of instance representations, and then proposed a novel two-stage prediction paradigm—Leveraging Label Distributions as Anchors (LDA). Different from traditional methods, LDA explicitly employs label distributions as anchors to guide the learning of instance representations, thereby optimizing these representations and enhancing prediction performance. Specifically, we first utilize a Vector Quantized Variational Auto-Encoder (VQ-VAE) [33] to discretize the continuous watch ratio distribution into representative anchors, effectively capturing the multi-peak characteristics within each duration group. Subsequently, the anchor vectors are directly incorporated into the recommendation model as parameters, and both instance representations and anchors are projected into a common latent space. Furthermore, LDA leverages Optimal Transport (OT) [34] to generate pseudo-labels for each instance based on the anchor distribution, enabling the model to acquire more structured coordinate information during training and to align closely with the anchors. Finally, the optimized instance representations are derived through a weighted integration of the anchor vectors, leading to more accurate and robust watch-time predictions. Our contributions can be summarized as follows:

- We propose the Leveraging Label Distributions as Anchors (LDA) method, which explicitly utilizes label distributions as anchors to guide the learning of instance representations, thereby improving watch-time prediction performance.
- We employ VQ-VAE to generate anchor vectors from the label distribution, explicitly capturing the multi-peak characteristics of the watch time distribution.
- We propose using Optimal Transport to generate pseudo-labels for each instance under the anchor distribution, thus obtaining more accurate instance representations.
- Extensive offline experiments on three datasets, along with large-scale online A/B testing on a short video platform, validate the significant advantages of the LDA method.

2 Related Work

2.1 Watch Time Prediction in Video Recommendation

Video recommender systems [5, 45] play a critical role in platforms like YouTube, TikTok, and Instagram, delivering personalized content to users. The watch-time metric has long been considered the core evaluation target in these systems, as it offers detailed insights into user engagement with videos. Early research employed time-weighted logistic regression to handle the semi-closed watch-time intervals during training [3]. Later studies revealed a strong relationship between watch-time distribution and video length, prompting the development of duration-based prediction methods [43, 47, 49]. For instance, methods like D2Q [43] and D^{2}Co [47] leverage duration bias to correct predictions, while DVR [49] approaches this challenge with a debiased metric and adversarial learning. ome works focus on modeling complex distribution patterns. TPM [20] constructs a watch-time prediction tree that allows probabilistic modeling at hierarchical quantiles, while CREAD [29] employs ordinal regression to discretize watch time and predict the likelihood of reaching each quantile. The CQE[19] method extends distribution modeling by formulating a quantile regression problem, enabling the model to capture the variability of watch-time across any quantile. SWAT [38] leverages a user-centric statistical framework with behavior-driven assumptions and bucketization techniques to model watch time.

2.2 Deep Clustering Method

Unsupervised clustering has traditionally relied on algorithms such as K-means [18], Gaussian Mixture Models (GMM) [27], and DBSCAN [12]. While effective for low-dimensional and well-separated data, these methods struggle with high-dimensional, non-linearly separable datasets. Deep learning has significantly advanced clustering by learning meaningful representations, with autoencoder-based and contrastive learning approaches being among the most common solutions [11, 17]. Contrastive learning method [1, 2]has recently been adopted to improve clustering by enforcing instance-wise discriminative features. While these methods improve latent representation quality, they rely on high-quality negative samples and have poorer training stability. Autoencoder methods can effectively compress data into a latent space, but they still need an extra traditional clustering step. Compared to the above, Vector Quantized Variational Autoencoders (VQ-VAEs) [32] have emerged as a pivotal approach in the domain of deep clustering for unsupervised learning tasks, effectively overcoming the limitations of previous clustering methodologies. These methods can learn discrete data embeddings for clustering without additional operations. They introduce discrete latent space by quantizing continuous latent vectors to codebooks, supported by techniques like the Straight-Through Estimator (STE) [39] and Gumbel-Softmax [10] for end-to-end training. The learned codebook act as cluster centroids [35, 48], enabling applications in conditional image generation [6, 26], multi-modal language modeling [14, 41], and recommender systems [21, 25].

3 Preliminaries

OT is a mathematical framework that transports a probability measure μ_1 onto another measure μ_2 with a minimum cost measured by some function $d(\cdot,\cdot)$., which has been extensively studied and applied in domains such as domain adaptation and recommender systems. In Kantorovich's relaxed formulation, OT seeks an optimal coupling $\mathbf{M} \in \Pi(\mu_1, \mu_2)$, defined as the joint probability distribution that minimizes a cost function over the space $\Pi(\mu_1, \mu_2)$ of probability distributions over $\mathbb{R}^2$ with marginals μ_1 and μ_2.

$$OT(\mu_1, \mu_2) = \inf_{\mathbf{M} \in \Pi(\mu_1, \mu_2)} \int_{\mathbb{R}^2} d(x_1, x_2)\, d\mathbf{M}(x_1, x_2), \tag{1}$$

where $d(x_1, x_2)$ is the cost of moving $x1$ to $x2$ (drawn from distributions μ_1 and μ_2, respectively). In the discrete versions of OT, i.e. when μ_1 and μ_2 are defined as empirical measures based on vectors in $\mathbb{R}^d$, $\Pi(\mu_1, \mu_2) = \{\mathbf{M} \in \mathbb{R}^{n \times m}_{\geq 0} | \mathbf{M}\mathbf{l}_m = \mu_1, \mathbf{M}^T\mathbf{l}_n = \mu_2\}$ denotes the polytope of matrices $\mathbf{M}$. Then, OT distance is formulated as the following optimization problem:

$$OT(\mu_1, \mu_2) = \min_{\mathbf{M} \in \Pi(\mu_1, \mu_2)} \langle \mathbf{M}, \mathbf{C} \rangle \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius dot product, $\mathbf{C} \in \mathbb{R}^{n \times m}_{\geq 0}$ is a cost matrix, representing the pairwise costs of transporting bin i to bin j. Directly solving this problem is computationally challenging due to the complexity of the constraints. Entropic regularization addresses this by introducing a regularized version of OT:

$$OT_\epsilon(\mu_1, \mu_2) = \min_{\mathbf{M} \in \Pi(\mu_1, \mu_2)} \langle \mathbf{M}, \mathbf{C} \rangle - \epsilon H(\mathbf{M}) \tag{3}$$

Here, $H(\mathbf{M}) = -\sum_{i,j} \mathbf{M}_{ij} \log(\mathbf{M}_{ij})$ is the entropy of the transport plan, promoting sparsity, and $\epsilon > 0$ is a regularization parameter that balances transport cost minimization with the smoothness of the solution. This regularization facilitates the use of the Sinkhorn-Knopp algorithm, which efficiently computes a transport plan by iteratively normalizing the rows and columns of an initial matrix to meet the marginal constraints.

4 Method

Section 4.1 presents the problem definition. Section 4.2 analyzes the motivation from both visualization and mathematical perspectives. Sections 4.3 and 4.4 provide a detailed content to our proposed two-stage LDA approach, and Section 4.5 discusses the time complexity. Fig3. shows the architecture of LDA.

4.1 Problem Definition

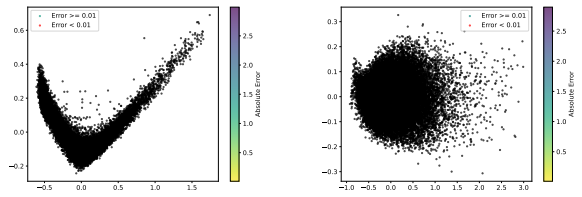
Let $\mathcal{U} = \{u_1, ..., u_{|\mathcal{U}|}\}$ and $\mathcal{V} = \{v_1, ..., v_{|\mathcal{V}|}\}$ denote the set of users and videos, respectively, where $|\mathcal{U}|$ is the number of users and $|\mathcal{V}|$ is the number of items. The user-item historical interactions are represented by $\mathcal{D} = \{(x_i, w_i) | x = (u, v), u \in \mathcal{U}, v \in \mathcal{V}\}_{i=1}^{N}$, where N is the number of samples and $w \in \mathbb{R}^*$ denotes the watch-time. The target is to learn a deep recommendation model $f(X; \theta_f)$ and a regressor $g(f(X; \theta_f); \theta_g)$ to predict the watch-time w of user u of video v, where θ_f and θ_g is the parameters of f and g, respectively.

4.2 Motivation

In this section, we investigate the impact of instance representations on watch-time prediction performance from two perspectives—intuitive visualization and rigorous theoretical proof—to provide comprehensive analysis of our work. First, from a visualization standpoint, we performed dimensionality reduction with PCA on instance representations for which the true values $y \in [0.0, 0.1]$ (Fig. 2(a))and $y \in [2.0, 2.1]$ (Fig. 2(b)). In visualizations, samples depicted in red correspond to those with a prediction error $|y - \hat{y}| \leq 0.01$, while darker hues in the *viridis colormap* indicate larger prediction errors. Observations reveal that, in both intervals, samples with lower prediction errors tend to cluster closely around a certain ideal center, whereas those with higher errors are more widely dispersed and lie farther from this center. This phenomenon intuitively demonstrates that when the instance representations are closer to their corresponding ideal centers, the model's predictions are more accurate; conversely, greater deviations from the ideal centers lead to significantly larger prediction errors. This observation provides a strong empirical basis for exploring methods to optimize instance representations in order to enhance prediction accuracy.

Second, from a theoretical perspective, we further establish the intrinsic relationship between the deviation of instance representations and prediction errors.

Proposition 4.1. *Let instance representation of a sample (x, y) be $f(x)$, ideal center defined as $\mu_y = \mathbb{E}[f(x) \mid y]$ for value y. Define the representation distance deviation as $D = \|f(x) - \mu_y\|$. Let the model's prediction be $\hat{y}$ and the prediction error be $\Delta_x = |y - \hat{y}|$. Δ_x is proportional to D, thereby indicating that the erroneous instance representation is the primary source of prediction errors.*



(a) 0≤y≤0.1 (b) 2.0≤y≤2.1

Figure 2: Dimensionality reduction visualization of sample representations with true values y in the ranges $[0, 0.1]$ and $[2.0, 2.1]$. Red indicates samples with a prediction error less than or equal to 0.01, while darker hues in the viridis colormap correspond to larger prediction errors.

proof: Consider a regression model with the output given by

$$\hat{y} = \mathrm{ReLU}(\mathbf{W}f(x) + b),$$

where $\mathbf{W} \in \mathbb{R}^{1\times d}$ and $b \in \mathbb{R}$. Assume that the true value can be expressed as

$$y = \mathrm{ReLU}(\mathbf{W}\mu_y + b) + \epsilon,$$

where the noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is independent of $f(x)$.

For the majority of samples, we assume that $\mathbf{W}f(x) + b \geq 0$ (i.e., they lie in the activation region). Then, we have

$$\hat{y} = \mathbf{W}f(x) + b \quad \text{and} \quad y = \mathbf{W}\mu_y + b + \epsilon.$$

Consequently, the prediction error is

$$\Delta_x = \left|\mathbf{W}\big(\mu_y - f(x)\big) + \epsilon\right|.$$

Squaring and taking expectations (noting that $\mathbb{E}[\epsilon] = 0$) yields

$$\mathbb{E}[\Delta_x^2] \approx \|\mathbf{W}(\mu_y - f(x))\|^2 + \sigma^2 = \|\mathbf{W}\|^2 \cdot \|f(x) - \mu_y\|^2 + \sigma^2.$$

Thus, aside from the constant noise term, the error is predominantly governed by $\|f(x) - \mu_y\|$.

4.3 Vector Quantize Label Distributions

The label Y is a one-dimensional sequence of length N. Transforming it into high-dimensional vectors while preserving its original distributional properties is a challenging task. We observe that the watch-ratio distributions Y in different video-duration buckets often exhibit pronounced aggregation and multi-peak characteristics; employing a single global modeling strategy may obscure the local structures within each bucket. Therefore, we partition Y into D sub-distributions $\mathbf{Y}_d$ based on video duration and learn each one independently, thereby more effectively retaining the multi-peak characteristics present in each bucket.

For each bucket, we sample L equal-length one-dimensional sequences $\{\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^L\}$, where each sequence serves as an approximate distribution of its bucket. To convert them into discrete and high-dimensional representations, we employ a Vector Quantized Variational Autoencoder (VQ-VAE) [33], training each bucket's sequences independently. The detailed procedure is as follows:

(1) Encoder: The sampled sequence $\mathbf{s}$ is fed into the encoder $E(\cdot; \Theta_E)$, yielding a latent representation $\mathbf{z}_e = E(\mathbf{s})$.

(2) Vector Quantization: From the codebook $\mathcal{A} \in \mathbb{R}^{K\times d}$, we select the anchor vector $\mathbf{a}_j$ that minimizes the Euclidean distance to $\mathbf{z}_e$, thereby discretizing $\mathbf{z}_e$ into an anchor representation $\mathbf{z}$:

$$\mathbf{z} = \arg\min_{\mathbf{a}_j \in \mathcal{A}} \|\mathbf{z}_e - \mathbf{a}_j\|.$$

Here, K denotes the number of anchor vectors, and each $\mathbf{a}_j \in \mathbb{R}^d$ encapsulates a particular distributional feature.

(3) Decoder: The discrete anchor $\mathbf{z}$ is then passed to the decoder $D(\cdot)$ to reconstruct the original sequence $\mathbf{s}$.

To address the gradient-blocking effect of the arg min operation, VQ-VAE adopt the Straight-Through Estimator (STE) [32] to approximate the backpropagation process. The training loss is formulated as follows:

$$\begin{aligned}\mathcal{L}_{VQ-VAE} = &\left\|\mathbf{s} - D\big(E(\mathbf{s}) + \mathrm{sg}\big[\mathbf{z} - E(\mathbf{s})\big]\big)\right\|_2^2 \\ &+ \left\|\mathrm{sg}\big[E(\mathbf{s})\big] - \mathbf{z}\right\|_2^2 + \beta\left\|E(\mathbf{s}) - \mathrm{sg}\big[\mathbf{z}\big]\right\|_2^2.\end{aligned} \tag{4}$$

Here, $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation. Through this process, we learn D sets of discrete anchors $\mathcal{A}_d = \{\mathbf{a}_1, ..., \mathbf{a}_K\} \in \mathbb{R}^{K*d}$ that capture multi-modal distributional characteristics, transforming the originally hard-to-model ultra-long one-dimensional vector into a high-dimensional discrete space.

4.4 Aligning Instance Representations to Anchors via Optimal Transport

First, for each sample x_i, we identify its duration bucket d and retrieve the corresponding anchor matrix $\mathcal{A}_d = \{\mathbf{a}_1, \ldots, \mathbf{a}_K\}$. Next, we apply the mapping function $\varphi(\cdot)$ to project both the instance representation $f(x_i)$ and the anchor vectors into a common space, resulting in

$$\mathbf{h}_i = \varphi\big(f(x_i)\big), \quad \mathbf{p}_k = \varphi\big(\mathbf{a}_k\big). \tag{5}$$

To further optimize instance representations and align them with the anchors, we employ Optimal Transport (OT) to generate pseudo-labels for each instance. These pseudo-labels provide alignment supervision during training, ensuring that $f(x_i)$ effectively matches the corresponding anchors in the common space.

4.4.1 Pseudo-labels generations. We begin by defining a cost matrix $\mathbf{C}$ that quantifies the dissimilarity between each representation $\mathbf{h}_i$ and anchor $\mathbf{p}_k$ based on cosine similarity:

$$\mathbf{C}_{i,k} = 1 - \mathrm{cosine}(\mathbf{h}_i, \mathbf{p}_k), \tag{6}$$

A smaller $\mathbf{C}_{i,k}$ indicates higher similarity between $\mathbf{h}_i$ and $\mathbf{p}_k$. Using the cost matrix $\mathbf{C}$, we apply OT to compute the transport plan $\mathbf{Q}^*$:

$$\mathbf{Q}^* = \mathrm{UOT}^{\kappa}(\mathbf{C}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2), \tag{7}$$

where $\boldsymbol{\alpha} = \delta_{\mathbf{h}_i}$ is a Dirac delta distribution representing the individual sample $\mathbf{h}_i$, and $\boldsymbol{\mu}_2 = \sum_{k=1}^{K} \delta_{\mathbf{p}_k}$ is the weight distribution over the K anchors $\mathbf{p}_k$. The regularization parameter κ controls the entropic smoothing of the transport plan. It ensures that each representation $\mathbf{h}_i$ is softly associated with multiple anchors based on their similarities.

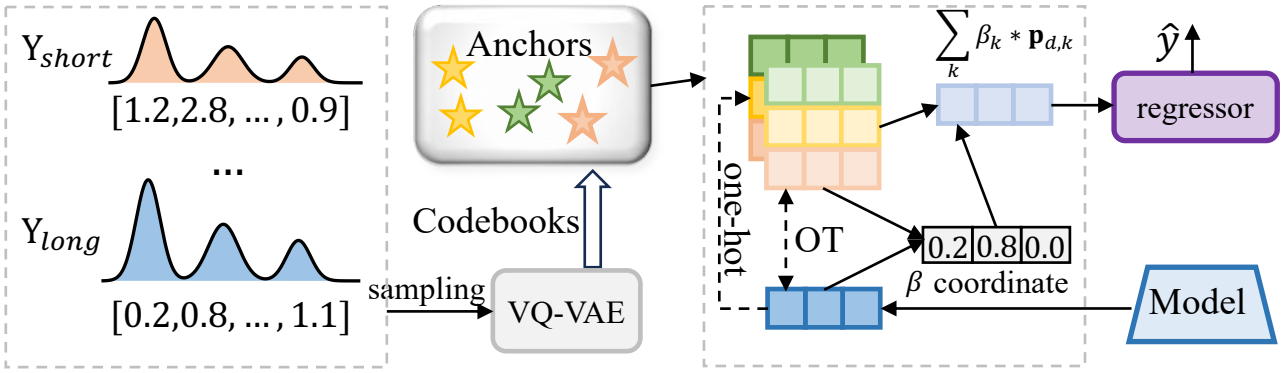


Figure 3: The framework of LDA, consisting of two stages: (a) anchors generations. and (b) alignment.

4.4.2 Alignment. Recall that each instance is projected into the common space as $\mathbf{h}_i$, and each anchor is represented as $\mathbf{p}_k$. We define $\beta_{i,k}$ as the coordinate of instance x_i with anchor $\mathbf{p}_k$:

$$\beta_{i,k} = \frac{\exp\left(\mathbf{h}_i^T * \mathbf{p}_k\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{h}_i^T * \mathbf{p}_j\right)}. \tag{8}$$

Next, the transport plan $\mathbf{Q}^*$ acts as pseudo-labels to guide the alignment between instance representations $\mathbf{h}_i$ and the anchor vectors $\mathbf{p}_k$. Specifically, we refine the representations by minimizing the following supervised loss:

$$\mathcal{L}_{sup} = -\frac{1}{K} \sum_{k=1}^{K} \mathbf{Q}^*_{i,k} * \log \beta_{i,k} \tag{9}$$

where τ is a temperature parameter controlling the sharpness of the distribution. This loss ensures that each $\mathbf{h}_i$ is re-represented in a way that reflects its relationships with the anchors $\mathbf{p}_k$. The supervised loss essentially pushes each $\mathbf{h}_i$ closer to the anchors it is most similar to (as indicated by $\mathbf{Q}^*$), while simultaneously discouraging similarity to unrelated anchors.

4.4.3 Prediction. After alignment, we use a regressor $g(\cdot)$ to predict the watch time with the linear combination of anchors. Specifically,

$$\hat{y} = g(\sum_{k=1}^{K} \beta_{i,k} * \mathbf{p}_k), \tag{10}$$

where $g(\cdot) = \text{ReLU}(\mathbf{W} \cdot + b)$. The mean squared error (MSE) loss is defined as:

$$\mathcal{L}_{mse} = (y_i - \hat{y}_i)^2 \tag{11}$$

We then formulate the total loss as:

$$\mathcal{L}_{total} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{mse} + \alpha \mathcal{L}_{sup}. \tag{12}$$

Here, α balances the importance of the supervised alignment loss and the primary prediction objective. This approach effectively captures the inherent structure of the label distribution, enhancing the model's robustness and leading to more accurate predictions.

4.5 Complexity analysis

In this section, we provide a detailed analysis of the computational overhead of LDA. LDA is a two-stage training scheme, and the anchor generation process using VQ-VAE is independent of the recommendation model's training, rendering its time overhead negligible. During the training phase of the recommendation model, the primary computational cost arises from using OT to generate pseudo-labels for each sample based on the anchor vectors corresponding to each bucket. According to the Sinkhorn algorithm, this process has a time complexity of $O(K \cdot \log(1/\epsilon))$. Additionally, computing the final prediction involves a linear combination of multiple anchors, which incurs a time complexity of $O(K)$. Therefore, the overall time complexity per sample during training is $O((1 + \log(1/\epsilon)) * K)$. In the testing phase, as the OT module is removed, the inference overhead reduces to $O(K)$.

5 Experiment

5.1 Dataset

We conduct our offline experiments on two publicly available datasets and one proprietary industrial dataset: **Wechat** (collected from the Wechat App), **KuaiRec** [7] (collected from the Kuaishou App), and **Indust** (collected from Kuaishou's production environment). Each dataset is randomly partitioned into training, validation, and test sets following a 6:2:2 ratio.

1) WeChat:This dataset was adopted in WeChat Big Data Challenge, which records the behavior of users on short videos in two weeks. It contains 20,000 users, 96,428 items, and 7,210,290 impression records. The user_id,device_id, video_id, author_id, duration_level and multi-model content feature vectors are used as inputs.

2) KuaiRec: KuaiRec is a real-world dataset collected from the recommendation logs of the video-sharing mobile app Kuaishou. It comprises 7,176 users, 10,728 distinct items, and 12,530,806 impression records, while its evaluation subset includes 1,411 users, 3,327 items, and 4,676,570 impression records. In our experiments, we select user_id, video_id, duration_level, author_id and music_id as our feature inputs.

3) Indust: The Indust dataset aggregates data from 0.5 million active users and 24 million items, encompassing almost 70 million interactions. We adopt the same feature inputs as those used in KuaiRec.

Table 1: Statistical information of datasets.

Dataset	#user	#item	#interaction	#duration
WeChat	20,000	96,428	7,210,290	5
KuaiRec	7,176	10,728	12,530,806	5
Indust	500,000	24,000,000	70,000,000	10

5.2 Setup

5.2.1 ***Baselines***. We adopt the following methods as baselines in our experiments:

- **VR(Value Regression).** This approach predicts watch time values directly by minimizing the mean squared error loss between the prediction and the ground truth.
- **WLR (Weighted Logistic Regression) [3].** This method employs a weighted logistic regression model and uses the learned odds as the predicted watch time.
- **OR(Ordinal Regression) [4].** This method employs ordinal regression techniques, emphasizing the relative order of watch times, making it suitable for fitting data to predict categorical levels of watch duration.
- **D2Q(Duration-Deconfounded Quantile) [43].**As a state-of-the-art approach approach in watch time prediction, this model addresses duration bias through backdoor adjustment and fits duration-dependent watch time quantiles using mean squared error.
- **TPM(Tree-based Progressive Model) [20].** This method employs a series of tree-structured classification tasks, taking into account ordinal ranks and prediction variance, and incorporates backdoor adjustments to reduce bias. It offers a detailed and comprehensive approach to enhancing watch time predictions in video recommendation systems.
- **CWM(Counterfactual Watch Model) [46].** This methods proposes to use counterfactual reasoning to mitigate duration bias.

5.2.2 ***Evaluation Metrics*** . To evaluate the performance of each model, we utilized two widely adopted metrics:

- **MAE(Mean Absolute Error)** is used to evaluate the average discrepancy between generated and real data; the calculation is as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \,. \tag{13}$$

- **XAUC**[42] is an extension of AUC designed for continuous values. For each pair of samples(i, j), if the predicted watch-time values $\hat{y_i}$ and $\hat{y_j}$ are in the same order as the ground truth, the score is 1; otherwise, the score is 0. We uniformly sample these pairs from the test set and average their scores to obtain XAUC. The formal definition is:

$$\mathrm{XAUC} = \frac{1}{|\mathcal{S}|}\sum_{(i,j)\in\mathcal{S}} \mathbb{I}\left[\left(\hat{y}_i > \hat{y}_j\right) = \left(y_i > y_j\right)\right] \tag{14}$$

where $\mathcal{S}$ represents the set of all sampled pairs, and $\mathbb{I}(\cdot)$ is the indicator function. Intuitively, XAUC measures how closely the ranking induced by the predicted watch times aligns with the ideal ranking. A higher XAUC value indicates better model performance.

Table 2: Watch-time prediction results on KuaiRec and WeChat.

Method	KuaiRec		Wechat		Indust	
	MAE	XAUC	MAE	XAUC	MAE	XAUC
VR	6.25	0.523	20.89	0.601	10.64	0.558
WLR	5.83	0.532	20.17	0.626	10.25	0.572
D2Q	5.21	0.563	19.05	0.633	9.88	0.589
OR	5.11	0.548	19.88	0.631	9.81	0.577
CWM	4.83	0.579	19.21	0.639	9.57	0.599
TPM	4.35	0.603	18.97	0.642	9.26	0.611
LDA	**3.22**	**0.611**	**18.42**	**0.655**	**8.72**	**0.628**

5.2.3 ***Implementation Details***. We set the embedding dimension for all features to 64. For TR and OR, both models are based on an MLP comprising three hidden layers, and ReLU[8] is used as the activation function. For other baseline methods, we follow the experimental design and tune the parameters to achieve optimal performance. On both datasets, we train all models using the Adam[13] optimizer with a batch size of 512. To prevent overfitting, we set the dropout[28] rate to 0.2 and adopt earlystopping[24] mechanism with a patience of 10. Specifically, we search for the learning rate in {1e-3, 1e-4, 1e-5}, tune α from 0.0 to 0.2 in increments of 0.05, and explore K in {5, 10, 15, 20, 25, 30}.

5.3 Performance Comparison

We evaluated the performance of the proposed LDA method against several baseline models on three real-world industrial datasets: KuaiRec, WeChat, and Indust. The results, as presented in Table 1, clearly demonstrate the superiority of LDA across all evaluation metrics, including **MAE** and **XAUC**.

Considering the model performance, LDA outperforms all other baselines, demonstrated by increased XAUC (i.e., better model performance) and reduced MAE (i.e., more reliability of prediction results). In contrast, the traditional method VR demonstrate the weakest performance across all datasets. This is likely due to its reliance on direct regression of watch time values, which fails to leverage the underlying distribution characteristics of the data. WLR, while showing improvements over VR, still falls short in comparison to other models, as it only captures the distribution of the odds and does not explicitly address duration biases or the complex relationships in the data. Similarly, OR provides a slight improvement by emphasizing the relative order of watch times but still lacks the robustness needed for more accurate predictions. D2Q enhances performance by addressing duration bias through backdoor adjustments, leading to better prediction accuracy. However, while D2Q outperforms the basic models like VR and WLR, it still does not reach the performance levels of TPM. TPM, with its tree-structured classification tasks, shows further improvements by capturing the dependency and uncertainty in the data. However, even TPM falls short of LDA's performance.

LDA's outstanding results are a direct consequence of its novel approach that optimizes instance representations by aligning them

Table 3: Ablation results of different modules of LDA.

Model	KuaiRec		Wechat		Indust	
	MAE	XAUC	MAE	XAUC	MAE	XAUC
LDA	**3.22**	**0.611**	**18.42**	**0.655**	**8.72**	**0.628**
w/o VQ-VAE	3.84	0.587	19.03	0.624	9.55	0.591
w/o OT	4.17	0.572	19.35	0.609	9.78	0.580
w/o LDA	6.25	0.523	20.89	0.601	10.64	0.558

with a distribution of representative anchors. This ensures that instances are accurately mapped to their corresponding positions in a shared space, where pseudo-labels generated via Optimal Transport (OT) align with anchor distributions. This design allows the model to effectively mitigate confusion in instance representations and enhances the robustness of predictions. By leveraging these anchor points, LDA captures the underlying distribution properties of user behavior more effectively than traditional regression-based methods. Its ability to utilize label distributions as anchors, combined with the power of VQ-VAE and OT-based alignment, positions LDA as the most accurate and robust method for watch time prediction in video recommendation systems.

5.4 Ablation Study

We conduct ablation studies on LDA to assess the individual impact of its components on model performance. Table 3 presents the XAUC and MAE metrics across various datasets, comparing performance with and without each component. Since LDA aims to enhance model performance by leveraging label distributions as anchors, in conjunction with the power of VQ-VAE and OT-based alignment, we can indirectly evaluate the contribution of each component by analyzing their effects on these metrics.

5.4.1 ***Effectiveness of different modules of LDA.*** We conducted further ablation studies to demonstrate the effectiveness of the components of LDA, with the relevant results presented in the Table 3. We compare LDA with its three variants. The results indicate that removing any single module results in a performance decline, proving that each component of LDA is crucial for enhancing the model's performance.

1) w/o VQ-VAE: It means that anchors are not generated from the continuous watch ratio distribution but are randomly initialized as parameters within the neural network. Removing VQ-VAE leads to a significant performance drop, highlighting the critical role of converting the continuous watch ratio distribution into representative anchors in effectively improving model performance.

2) w/o OT: In this variant, OT is removed. The drop in metrics is particularly notable, indicating that OT helps better align the distributions of instances and anchor, thereby enhancing predictive ability.

3) w/o LDA: Without LDA, the approach degenerates into Value Regression.

5.4.2 ***Different anchor generation methods.*** To further validate the effectiveness of using VQ-VAE to generate representative anchors, we compared two different generation strategies: KMeans, and random generation. As shown in Table 4, the performance of KMeans slightly declines, indicating that the representative anchors from VQ-VAE exhibit better clustering structures, making it easier for instance representations to align. KMeans generates anchors by directly clustering instance representations, but its performance suffers due to greater sensitivity to noise and potential errors. The random method performs the worst because it fails to provide a representative reference for alignment, which negatively impacts the model's prediction performance. Additionally, we observe that models using the random and K-means methods converge more slowly compared to those using VQ-VAE. This is because the anchors constructed through VQ-VAE contain posterior information about the label distributions, which guides the CPF learning process, allowing it to search for the global optimum more quickly and accurately.

Table 4: Ablation study on different anchor generation.

Anchor generation	KuaiRec		Wechat		Indust	
	MAE	XAUC	MAE	XAUC	MAE	XAUC
VQ-VAE	**3.22**	**0.611**	**18.42**	**0.655**	**8.72**	**0.628**
Kmeans	3.81	0.592	18.99	0.626	9.49	0.595
Random	3.84	0.587	19.03	0.624	9.55	0.591

5.4.3 ***Different alignment methods.*** We also compared different alignment methods, as shown in Table 5. OT achieves the best performance across all datasets, which demonstrates the effectiveness of OT-based alignment. This approach considers global distribution alignment, offering strong robustness and interpretability.

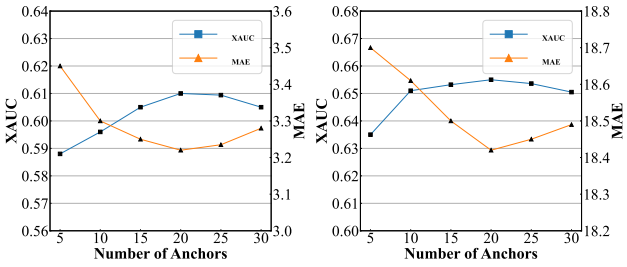
To validate the effectiveness of OT, we use the L2 distance to assign instances to relevant anchors. Specifically, we calculate the L2 distance between instances and anchors, and then apply the negative L2 distance in a softmax function to obtain the corresponding anchor weights. By leveraging these weights, we employ weighted linear combinations of the anchors to represent the instances. This approach directly aligns two representations, focusing on point-wise alignment without considering the global distribution. It makes it susceptible to the influence of outliers. As shown in Table 5, this limitation restricts the performance of L2 distance. Additionally, w/o alignment also sees a significant performance decline in both metrics across all the datasets, as it directly uses the linear combination of anchors to represent the instances. In conclusion, OT-based alignment more effectively enhances model performance.

5.4.4 ***Impact of the number of anchors K.*** In Figure 4, we show the impact of the number of anchors K on model performance. As the number of anchors increases, performance improves accordingly. However, defining too many anchors leads to slight fluctuations in performance, which may be caused by the introduction of noise. Given the trade-off between the MAE and XAUC, we adopt $K = 20$ in all our experiments.

5.4.5 ***Results on Different Duration Buckets.*** We further investigate the performance of our model by evaluating it across various duration buckets, without any direct comparisons to other methods. Table 6 presents the results for short, medium, and long-duration buckets. Overall, our model demonstrates steady performance across different buckets, indicating that it can effectively

Table 5: Ablation results on different alignment methods.

Model	KuaiRec		Wechat		Indust	
	MAE	XAUC	MAE	XAUC	MAE	XAUC
OT	**3.22**	**0.611**	**18.42**	**0.655**	**8.72**	**0.628**
L2 distance	3.58	0.608	18.79	0.650	9.02	0.615
w/o alignment	4.31	0.599	19.55	0.635	9.91	0.586



(a) KuaiRec (b) Wechat

Figure 4: Impact of the number of anchors K.

handle the variation in watch-time patterns associated with different video lengths.

Table 6: Results on different duration buckets.

Duration bucket	KuaiRec		Wechat	
	MAE	XAUC	MAE	XAUC
0	2.58	0.615	6.34	0.657
1	3.06	0.602	9.78	0.648
2	3.28	0.588	14.90	0.646
3	5.59	0.592	23.68	0.625
4	7.62	0.557	37.88	0.628

5.5 Online A/B Testing

To validate the efficacy of LDA, we deploy it within the feed recommendation pages of two applications (Main and Light) of KuaiShou for online A/B test. Specifically, our recommendation system is a cascaded system comprising four stages: recall, pre-ranking, ranking, and re-ranking. We conduct A/B test in the re-ranking stage, using D2Q as the baseline model. As shown in Figure 5, we add a pre-generated anchor $\mathcal{A} \in R^{D*K*d}$ in the final fully connected layer, where D represents the number of buckets, K is the number of anchors in each bucket, and d denotes the hidden dimension. We evaluate the performance of LDA within a practical application context of Kuaishou, referencing primarily the subsequent four metrics: (1) #Watch Time: a crucial measure reflecting user engagement and satisfaction. (2) #Average app usage time: the average duration users spend on the app, serving as a key indicator of overall platform engagement.(3) #Like: a measure of user interaction, reflecting how much users appreciate or engage with the content.(4) #Follow: the number of new followers gained, indicating the effectiveness of recommendations in driving long-term user engagement.

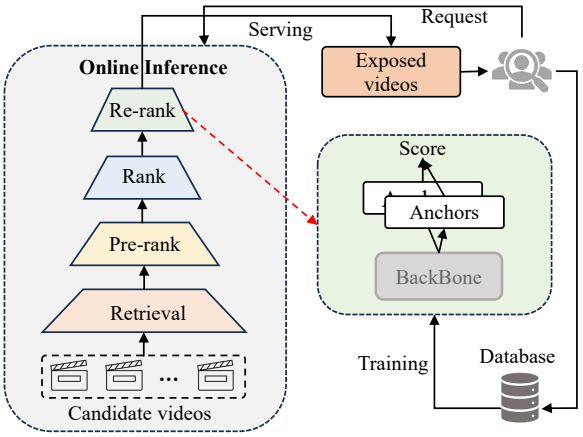


Figure 5: The development of LDA.

Table 7: Results of online A/B test.

APP	watch time	app usage	like	follow
Main	+0.039%	+0.021%	+0.401%	+0.108%
Light	+0.122%	+0.098%	+0.379%	+0.147%

The A/B test spanned 15 consecutive days, with the average performance on the four aforementioned metrics tabulated in Table 7, allowing us to formulate the ensuing conclusions. Initially, LDA exhibits a noticeable increase in Watch Time, highlighting the model's effectiveness in enhancing user engagement and retention. Following this, the improvement in App Usage further demonstrates the model's positive impact on user activity within the app. Additionally, the uplift in likes and follows confirms that the model contributes to fostering user interaction and expanding the user base.

In conclusion, LDA improves all four metrics compared to the baseline D2Q, proving that LDA can effectively achieve accurate watch-time prediction and better recommendation.

6 Conclusion

In this work, we proposed LDA, a novel paradigm for watch time prediction that leverages label distributions as anchors to optimize instance representations. By employing VQ-VAE to derive representative anchors from multi-peak watch ratio distributions and utilizing Optimal Transport to generate aligned pseudo-labels, our method enhances both accuracy and robustness. Extensive experiments on two datasets and large-scale online A/B testing on a platform with over 300 million DAUs confirm the effectiveness of our approach, highlighting its potential to improve video recommendation systems.

Acknowledgement

This research is partially sponsored by funding from Yonghua Foundation.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*. 132–149.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems* 33 (2020), 9912–9924.

[3] 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.

[4] Koby Crammer and Yoram Singer. 2001. Pranking with ranking. *Advances in neural information processing systems* 14 (2001).

[5] 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.

[6] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12873–12883.

[7] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-Observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. 540–550. https://doi.org/10.1145/3511808.3557220

[8] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 315–323.

[9] Xudong Gong, Qinlin Feng, Yuan Zhang, Jiangling Qin, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. 2022. Real-time short video recommendation on mobile devices. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 3103–3112.

[10] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[11] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2016. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148* (2016).

[12] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. 2014. DBSCAN: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*. IEEE, 232–238.

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125* (2023).

[15] Fan Li, Xu Si, Shisong Tang, Dingmin Wang, Kunyan Han, Bing Han, Guorui Zhou, Yang Song, and Hechang Chen. 2024. Contextual Distillation Model for Diversified Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5307–5316.

[16] Meng Li and Haochen Sui. 2025. Causal Recommendation via Machine Unlearning with a Few Unbiased Data. In *AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques*. https://openreview.net/forum?id=7btnaN4Std

[17] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2021. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 8547–8555.

[18] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition* 36, 2 (2003), 451–461.

[19] Chengzhi Lin, Shuchang Liu, Chuyuan Wang, and Yongqi Liu. 2024. Conditional Quantile Estimation for Uncertain Watch Time in Short-Video Recommendation. *arXiv preprint arXiv:2407.12223* (2024).

[20] Xiao Lin, Xiaokai Chen, Linfeng Song, Jingwei Liu, Biao Li, and Peng Jiang. 2023. Tree based progressive regression model for watch-time prediction in short-video recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4497–4506.

[21] Qijiong Liu, Xiaoyu Dong, Jiaren Xiao, Nuo Chen, Hengchang Hu, Jieming Zhu, Chenxu Zhu, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Vector Quantization for Recommender Systems: A Review and Outlook. *arXiv preprint arXiv:2405.03110* (2024).

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

[23] Hongxu Ma, Kai Tian, Tao Zhang, Xuefeng Zhang, Chunjie Chen, Han Li, Jihong Guan, and Shuigeng Zhou. 2024. Generative Regression Based Watch Time Prediction for Video Recommendation: Model and Performance. *arXiv preprint arXiv:2412.20211* (2024).

[24] Lutz Prechelt. 2002. Early stopping-but when? In *Neural Networks: Tricks of the trade*. Springer, 55–69.

[25] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).

[26] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[27] Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741, 659-663 (2009).

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[29] Jie Sun, Zhaoying Ding, Xiaoshuang Chen, Qi Chen, Yincheng Wang, Kaiqiao Zhan, and Ben Wang. 2024. CREAD: A Classification-Restoration Framework with Error Adaptive Discretization for Watch Time Prediction in Video Recommender Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9027–9034.

[30] Shisong Tang, Qing Li, Xiaoteng Ma, Ci Gao, Dingmin Wang, Yong Jiang, Qian Ma, Aoyang Zhang, and Hechang Chen. 2022. Knowledge-based temporal fusion network for interpretable online video popularity prediction. In *Proceedings of the ACM Web Conference 2022*. 2879–2887.

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

[32] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

[33] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural Discrete Representation Learning. *CoRR* abs/1711.00937 (2017). arXiv:1711.00937 http://arxiv.org/abs/1711.00937

[34] Cédric Villani et al. 2009. *Optimal transport: old and new*. Vol. 338. Springer.

[35] Hanwei Wu and Markus Flierl. 2020. Vector quantization-based regularization for autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6380–6387.

[36] Yuntian Wu, Yuntian Yang, Jiabao Sean Xiao, Chuan Zhou, Haochen Sui, and Haoxuan Li. 2024. Invariant Spatiotemporal Representation Learning for Cross-patient Seizure Classification. In *The First Workshop on NeuroAI @ NeurIPS2024*. https://openreview.net/forum?id=Ex6wAivo7G

[37] Yafei Xiang, Shuning Huo, Yichao Wu, Yulu Gong, and Mengran Zhu. 2024. Integrating AI for Enhanced Exploration of Video Recommendation Algorithm via Improved Collaborative Filtering. *Journal of Theory and Practice of Engineering Science* 4, 02 (2024), 83–90.

[38] Shentao Yang, Haichuan Yang, Linna Du, Adithya Ganesh, Bo Peng, Boying Liu, Serena Li, and Ji Liu. 2024. SWaT: Statistical Modeling of Video Watch Time through User Behavior Analysis. *arXiv preprint arXiv:2408.07759* (2024).

[39] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. 2019. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662* (2019).

[40] Savvas Zannettou, Olivia Nemes-Nemeth, Oshrat Ayalon, Angelica Goetzen, Krishna P Gummadi, Elissa M Redmiles, and Franziska Roesner. 2024. Analyzing User Engagement with TikTok's Short Format Video Recommendations using Data Donations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–16.

[41] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. 2024. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226* (2024).

[42] Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, Dong Zheng, and Peng Jiang. 2022. Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation. arXiv:2206.06003 [cs.IR] https://arxiv.org/abs/2206.06003

[43] Ruohan Zhan, Changhua Pei, Qiang Su, Jianfeng Wen, Xueliang Wang, Guanyu Mu, Dong Zheng, Peng Jiang, and Kun Gai. 2022. Deconfounding duration bias in watch-time prediction for video recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4472–4481.

[44] Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. 2024. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[45] Min Zhang and Yiqun Liu. 2021. A commentary of TikTok recommendation algorithms in MIT Technology Review 2021. *Fundamental Research* 1, 6 (2021), 846–847.

[46] Haiyuan Zhao, Guohao Cai, Jieming Zhu, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. 2024. Counteracting Duration Bias in Video Recommendation via Counterfactual Watch Time. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1–12.

[47] Haiyuan Zhao, Lei Zhang, Jun Xu, Guohao Cai, Zhenhua Dong, and Ji-Rong Wen. 2023. Uncovering User Interest from Biased and Noised Watch Time in Video Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 528–539.

[48] Chuanxia Zheng and Andrea Vedaldi. 2023. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22798–22807.

[49] Yu Zheng, Chen Gao, Jingtao Ding, Lingling Yi, Depeng Jin, Yong Li, and Meng Wang. 2022. Dvr: Micro-video recommendation optimizing watch-time-gain under duration bias. In *Proceedings of the 30th ACM International Conference on Multimedia*. 334–345.