UniMixer: Unifying Context and Latent Distribution Tokens for Generative Watch Time Prediction

1st Yichen Huang
Software Engineering Institute, East China Normal University
Shanghai, China
51275902070@stu.ecnu.edu.cn

2nd Jing Liu*
Software Engineering Institute, East China Normal University
Shanghai, China
jliu@sei.ecnu.edu.cn
*Corresponding author

3rd Li Han
Software Engineering Institute, East China Normal University
Shanghai, China
hanli@sei.ecnu.edu.cn

4th Guanyu Lin
Software Engineering Institute, East China Normal University
Shanghai, China
51275902135@stu.ecnu.edu.cn

5th Erxue Zhou
Software Engineering Institute, East China Normal University
Shanghai, China
51275902066@stu.ecnu.edu.cn

6th Zhengkang Zhou
Software Engineering Institute, East China Normal University
Shanghai, China
51275902071@stu.ecnu.edu.cn

***Abstract*—Accurate watch time prediction is essential for optimizing content delivery in streaming short-video platforms, yet it remains challenging due to the inherently complex and multimodal nature of watch time distributions. Through systematic analysis of real-world industrial data, we identify two critical challenges: (1) unobservability of latent multi-duration distributions, where watch time emerges from the entangled interaction between users' latent preferences and videos' content structures, and (2) inter-distribution cross-dependency, where different distribution components dynamically overlap and modulate each other along the temporal axis. To address these challenges, we assume that watch time follows an Exponential-Log-Normal mixture (ELN) distribution, where the exponential component captures quick-skip behaviors and the log-normal components characterize diverse long-tail engagement patterns. Accordingly, we propose UniMixer, a linear-complexity architecture that parameterizes each distribution component as a learnable token and achieves deep interaction between contextual features and distribution prototypes through a unified token-mixing mechanism. Extensive experiments on two large-scale public datasets, KuaiRec and WeChat, demonstrate that UniMixer achieves state-of-the-art performance on both ranking (XAUC) and regression (MAE) metrics. Remarkably, UniMixer maintains computational efficiency suitable for real-time industrial deployment while exhibiting superior distribution fitting capability across diverse user-video interaction patterns. Our code is available at https://github.com/Lambert-vsziii/UniMixer.**

***Index Terms*—watch time prediction, mixture distribution, video recommendation, distribution modeling, neural network applications**

I. Introduction

The rapid advancement of mobile internet has witnessed the explosive growth of short-video platforms such as TikTok, YouTube Shorts, and Kuaishou, which have fundamentally reshaped how users consume digital content [1]–[7]. Unlike traditional e-commerce or news recommendation scenarios that rely heavily on explicit user actions such as clicks, these modern platforms typically employ immersive, full-screen streaming mechanisms where content auto-plays in a continuous swipe loop [8], [9]. Under this passive interaction paradigm, conventional binary metrics such as Click-Through Rate (CTR) are no longer sufficient to capture the nuances of user preferences. Instead, watch time (or dwell time) has emerged as the de facto gold standard for quantifying user satisfaction and engagement depth [10]–[12]. Consequently, the ability to accurately model and predict watch time is critical for recommendation systems to optimize content distribution, ensuring that delivered videos precisely align with users' latent interests and thereby fostering long-term retention.

Watch time prediction is typically formulated as a regression task over continuous values; however, its labels often exhibit complex characteristics such as strong skewness, heavy tails, and even multimodality, rendering direct regression more challenging in terms of optimization and fitting. Therefore, adopting more realistic distributional assumptions about the generative process of watch time and modeling accordingly can often significantly reduce learning difficulty and improve regression accuracy [13]–[18]. Nevertheless, existing research generally lacks a unified and explicit distributional assumption along with mechanistic attribution analysis for watch time distributions. To bridge this knowledge gap, we conduct an empirical study on real-world watch time distributions from industrial-scale datasets. As illustrated in Figures 1 and 2, we analyze and compare watch time distributions ranging from single-video distributions to the complex mixture distributions induced by user-video conditional distributions:

The watch time distribution of a single video should be viewed as being jointly generated by multiple interrelated latent distributions, which fuse together to produce the real-world watch time distribution. Figure 1 provides a representative illustration of this phenomenon: at the individual video level, watch time distributions corresponding to different content formats often exhibit pronounced multimodal characteristics, with substantial variation across videos. Taking

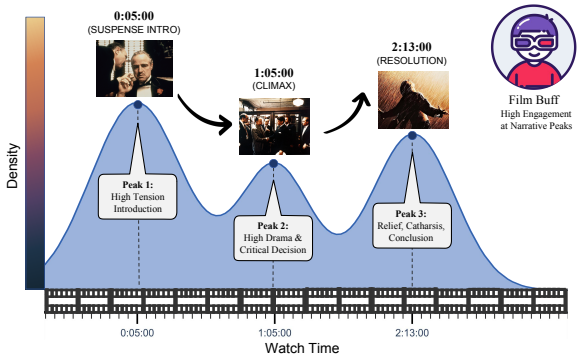

Fig. 1. Watch time distribution for a single video exhibiting multimodal characteristics. The distribution shows three distinct peaks corresponding to key narrative moments: (1) high tension introduction, (2) climax with high drama and critical decisions, and (3) resolution with relief and catharsis. This illustrates how video content structure generates multiple interrelated latent distributions.

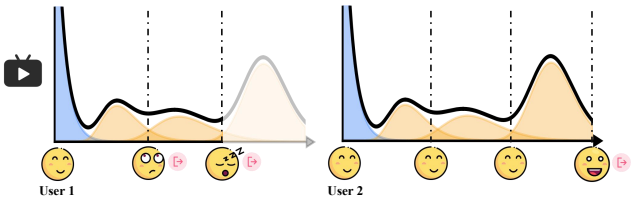

Fig. 2. User-video conditional watch time distributions. Different users exhibit distinct watch time patterns for the same video due to varying preferences, resulting in mixture distributions shaped by the coupling of video-side and user-side complexity.

narrative-driven film clips as an example, their segmented structure typically creates common exit points at several key plot transitions, causing watch time to cluster into multiple peaks around these nodes. Meanwhile, some users become strongly engaged by certain segments and continue watching until subsequent climaxes or endings, thereby forming additional peaks in longer duration regions. Furthermore, these peaks are not mutually independent: the anticipation and immersion established by preceding segments significantly influence whether users proceed to subsequent plot highlights, causing the latent distribution components corresponding to different peaks to overlap and exhibit conditional coupling along the temporal axis, ultimately shaping the complex mixture distribution patterns shown in Figure 1.

Building upon this foundation, we further examine the generative mechanism of watch time distributions from the user-video conditional perspective. Figure 2 provides an intuitive illustration: given that a single video already exhibits a complex multi-latent-distribution structure, the inherent complexity of different users' preference distributions further modulates these distribution components, thereby shaping even more diversified watch time patterns. Specifically, variations among users in terms of genre interests, plot focus, and immersion tendencies lead to differential response intensities toward different latent distribution components within the same video. For instance, some users prefer rapid information acquisition and tend to exit at early stages, while others are more likely to be captivated by mid-to-late plot developments and exhibit prolonged viewing behavior. Consequently, the multi-latent-distribution structure on the video side couples with the complex preference distributions on the user side during the conditioning process, causing the weights and combination patterns of distribution components to vary across users, ultimately forming real-world watch time distributions jointly determined by both video complexity and user complexity.

Based on the above analysis, we distill two key challenges facing generative watch time modeling:

Unobservability of Latent Multi-Duration Distributions. Watch time fundamentally manifests as the result of interactions between users' latent distributions and videos' inherent content distributions. However, existing methods struggle to capture such fine-grained signals and thus fail to fit the true playback distributions.

Inter-Distribution Cross-Dependency. The shape of the final watch time distribution is dynamically modulated by the specific user-video context. Different latent distribution components often overlap, intersect, and exert synergistic influences along the temporal axis, causing the overall distribution to exhibit highly nonlinear and structurally complex mixture patterns.

Existing methods typically mitigate the stochasticity of watch time through label normalization or task transformation. However, the former risks losing absolute temporal information [13]–[15], while the latter is constrained by discretization errors [16], [17]. More critically, mainstream paradigms rely on point estimation (i.e., conditional expectation), which fails to capture the inherently cross-interactive nature between user interests and video-user dynamics. Although recent works attempt to address duration bias through quantile modeling or standardization, they often overlook noisy viewing behaviors and depend on rigid distributional assumptions that may fail in practice. These limitations underscore the necessity of transitioning toward a multi-distribution mixture paradigm—one that explicitly models the complete conditional distribution and leverages adaptive user-video interactions to disentangle complex real-world distributions.

To address these challenges, and inspired by previous works in other fields [19]–[21], we reformulate watch time prediction through a generative multi-distribution perspective and propose **UniMixer** (**Uni**fying Context and Latent Distribution Tokens in a Token-**Mixer** Framework). Departing from the conventional point regression paradigm, we treat watch time as a random variable jointly generated by an exponential distribution and a log-normal mixture distribution, instantiating each distribution as learnable distribution tokens. Subsequently, UniMixer establishes deep interactions between contextual information and distribution prototypes, explicitly capturing the high-order dependencies and dynamic coupling among different viewing patterns, and ultimately outputs the distribution parameters and mixture weights to generate the

target watch time distribution.

Through extensive experiments on large-scale industrial datasets, we demonstrate that UniMixer achieves new state-of-the-art performance on both XAUC (measuring ranking capability) and MAE (measuring regression accuracy) metrics. This empirical success is primarily attributed to our proposed generative multi-distribution fitting paradigm, which enables the model to capture the complex multimodal shapes of watch time more accurately than conventional discriminative baselines. Furthermore, we emphasize the favorable trade-off between accuracy and efficiency: the proposed linear-complexity UniMixer architecture significantly reduces computational overhead while maintaining modeling capacity, making it well-suited for high-throughput real-world recommendation scenarios.

In summary, the main contributions of this paper are as follows:

- We propose a generative multi-distribution paradigm that models real-world watch time distributions as an Exponential-Log-Normal mixture (ELN), explicitly decoupling stochastic noise from long-tail engagement patterns, thereby addressing the challenges of unobservable latent viewing patterns and hard-to-characterize multimodal distributions.
- We propose UniMixer, a linear-complexity architecture that unifies contextual features and latent distribution prototypes as learnable tokens. Through deep token-mixing mechanisms, it achieves an efficient yet expressive multi-distribution parameter generation method suitable for constructing predictions under the ELN paradigm.
- We demonstrate through extensive experiments on industrial-scale datasets that UniMixer achieves state-of-the-art performance on both ranking (XAUC) and regression (MAE) metrics, while attaining an outstanding balance between model performance and computational efficiency.

II. Related Work

Watch-time Prediction Watch-time prediction is a fundamental task in video recommendation, serving as the most direct proxy for user engagement. However, traditional approaches treat watch time as a scalar target and train models with standard regression losses, implicitly imposing assumptions on both label and error distributions. Early methods relied on mean squared error (MSE), which assumes Gaussian residuals and can be mismatched with real-world data. In practice, watch-time distributions are typically highly skewed and long-tailed. Industrial models such as weighted logistic regression (WLR) [5] use watch time as impression weights, improving YouTube recommendations. Nevertheless, such formulations remain ill-suited to short-video platforms, where watch time exhibits more pronounced long-tailed behavior and video lengths vary substantially.

Duration Debiasing In recent years, a series of debiasing methods targeting duration bias have emerged, aiming to extract unbiased user interests from noisy observations. For instance, D2Q [13] mitigates duration bias via backdoor adjustment and by modeling watch-time statistics across different duration groups. However, D2Q relies on a fixed bucketization strategy over durations and overlooks the distributional imbalance of watch time in tail samples, resulting in substantially lower predictive accuracy on the tail than on the head. Beyond correcting duration bias, D^2CO [14] further targets users' exploratory behaviors (noisy watching) by applying a Gaussian Mixture Model (GMM) to adjust model training. Meanwhile, the CWM [15] model attributes duration bias to the truncation of a user's theoretical maximum watch time by video length, and introduces counterfactual watch time to recover the engagement signal truncated by duration. However, CWM's truncation mechanism ignores behaviors such as timeline seeking and repeated watching, which may lead to missing information in the estimated distribution.

Regression by Classification Another line of work reformulates the regression problem as a series of classification tasks, preserving the ordinal relationships among watch-time levels and achieving significant performance gains. TPM [16] discretizes watch time into ordered intervals and models them with a hierarchical tree of binary classifiers. Given the highly imbalanced nature of watch-time data, CREAD [17] introduces an error-adaptive discretization technique to construct dynamic time intervals, striking a balance between classification accuracy and reconstruction fidelity. CQE [22] and AlignPxtr [23] extend this paradigm by estimating conditional quantiles from a single observed watch-time value for each user–video pair. These methods demonstrate the effectiveness of classification-based formulations, yet the optimal strategy for distributional modeling remains underexplored.

Distributional and Generative Methods Finally, recent studies have revisited watch-time modeling from a distributional and generative perspective, offering new directions for this field. EGMN [19] introduces an Exponential–Gaussian Mixture framework, capturing both the long-tailed and multimodal nature of the watch-time distribution within a unified probabilistic formulation; however, its components are mainly aggregated only at the final weighted-combination stage, making it difficult to explicitly condition and compose different watching modes under context, and thus limiting its ability to model higher-order dependencies. Meanwhile, GR [24] proposes a Generative Regression framework that jointly models watch time together with other continuous variables; although GR recovers continuous values via token accumulation, its objective still leans toward reconstructing a single scalar observation, and therefore struggles to directly characterize variance shifts, uncertainty, and potential multimodality in the conditional distribution.

III. Methodology

A. Problem Formulation

Let $\mathcal{U}$ and $\mathcal{V}$ denote the user and video sets. For a user-video pair (u, v), we encode it as $\mathbf{x}_{u,v} = \phi(u, v) \in \mathbb{R}^n$. The

training data consists of historical interactions:

$$\mathcal{D} = \{(\mathbf{x}_i, w_i, d_i)\}_{i=1}^{N} \tag{1}$$

where $w_i \in \mathbb{R}^+$ is the observed watch time and d_i is the video duration.

We model the observed watch time as a superposition of multiple components:

$$w_{u,v} = \underbrace{b_{u,v}}_{\text{bias}} + \underbrace{p_{u,v}}_{\text{preference}} + \underbrace{\epsilon_{u,v}}_{\text{noise}} \tag{2}$$

where $b_{u,v}$ captures systematic bias from confounding factors (e.g., duration, category), $p_{u,v}$ represents the user's true interest, and $\epsilon_{u,v}$ is random noise.

Our goal is to learn a scoring function $f_\theta : \mathbb{R}^n \to \mathbb{R}$ that predicts the unobservable true preference p. The ideal objective is:

$$\theta^* = \arg\min_\theta \ \mathbb{E}_{(\mathbf{x},p)\sim\mathcal{D}} \left[\mathcal{L}\left(f_\theta(\mathbf{x}), p\right)\right] \tag{3}$$

Since p is unobservable, practitioners typically optimize:

$$\hat{\theta} = \arg\min_\theta \ \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},w)\in\mathcal{D}} \mathcal{L}\left(f_\theta(\mathbf{x}), w\right) \tag{4}$$

However, a gap exists between $\hat{\theta}$ and θ^* due to bias and noise in w. This paper aims to design a distributional modeling mechanism that uncovers users' true interests from noisy watch time.

B. Modeling Assumptions for Watch Time Distribution

Inspired by the typical patterns exhibited by watch time distributions, we assume that user watch time t follows a mixture model composed of an exponential distribution and a log-normal mixture distribution, with the probability density function:

$$p(t) = \omega_0 f_{\text{exp}}(t \mid \lambda) + \sum_{k=1}^{K} \omega_k f_{\text{logn}}(t \mid \mu_k, \sigma_k^2) \tag{5}$$

where $f_{\text{exp}}(t \mid \lambda) = \lambda e^{-\lambda t}$ is the density function of the exponential distribution with rate parameter λ, $f_{\text{logn}}(t \mid \mu, \sigma^2) = \frac{1}{t\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log t - \mu)^2}{2\sigma^2}\right)$ is the density function of the log-normal distribution with parameters (μ, σ^2), and ω_k represents the mixture weight of the k-th component, satisfying the normalization constraint $\sum_{k=0}^{K} \omega_k = 1$.

The rationale for this distribution choice is as follows:

Exponential Component. At the coarse-grained level, user "quick-skip" behavior forms a prominent left peak in the duration distribution. The exponential distribution, with its memoryless property and characteristic of concentrated probability mass near zero, can effectively capture this phenomenon. This component is primarily responsible for modeling users' rapid exit behavior when they are not interested in the video [25].

Log-Normal Components. At the fine-grained level, interactions between different user groups and diverse video content produce complex long-tail consumption patterns. Compared to the Gaussian distribution, the log-normal [26] distribution offers several advantages: (1) its support is naturally defined on the positive real domain $\mathbb{R}^+$, which better aligns with the non-negative physical constraint of watch time; (2) its right-skewed shape can better capture the long-tail characteristics of watch time; (3) it exhibits normality in the log space, making parameter estimation more stable. Through the mixture of K log-normal components, the model can flexibly fit the multi-modal distribution patterns jointly determined by user preferences and video characteristics.

C. Our Approach: UniMixer

This section proposes the UniMixer model, whose core innovation lies in: parameterizing each component of the mixture distribution as a learnable token representation, and achieving deep interaction between components and context through a unified token mixing mechanism. This design enables the model to dynamically capture dependencies between different distribution components, thereby more accurately modeling complex watch time distributions. The framework diagram is shown in Fig. 3.

1) Hidden Representation Encoder: For each user-video pair, we collect features from multiple sources and construct the input vector. Specifically, the feature sources include:

- **User features**: demographic information, historical behavior patterns, interaction statistics, etc.
- **Video features**: content attributes, duration, category, creator information, etc.
- **Context features**: time of day, weekday/weekend patterns, device type, etc.

All features are processed through embedding layers and concatenated into a feature vector $\mathbf{x}$, which is then fed into a shared feature encoding backbone network to obtain the hidden representation:

$$\mathbf{h} = g_{\text{backbone}}(\mathbf{x}) \in \mathbb{R}^d \tag{6}$$

where g_{backbone} can be instantiated as any feature encoder suitable for recommendation scenarios (such as DCN [27], DIN [28], Transformer [29], etc.). It is worth emphasizing that UniMixer, as a modeling paradigm decoupled from the backbone network, can flexibly adapt to different feature engineering schemes and model architectures, which facilitates its deployment in industrial systems.

2) Distribution Tokenization and UniMixer Fusion Module: The core idea of UniMixer is treating each component of the mixture distribution as an independent semantic unit and parameterizing it as a learnable token representation. Through this design, the distribution modeling problem is transformed into a representation learning and interaction modeling problem for token sequences, thereby leveraging mature sequence modeling techniques to enhance information flow between distribution components.

Specifically, we define learnable component tokens $\{\mathbf{z}_k\}_{k=1}^{K} \in \mathbb{R}^{K\times d}$ for the K log-normal components. These tokens are continuously updated through gradient optimization during training, gradually learning distribution semantics that can characterize different viewing patterns. Meanwhile, we transform the hidden representation $\mathbf{h}$ through a projection

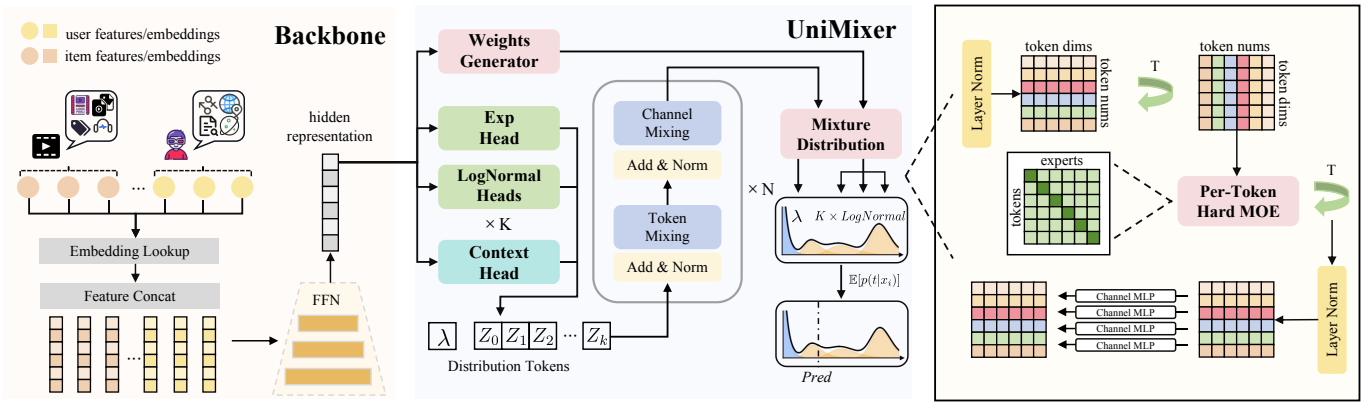

Fig. 3. Overview of the UniMixer framework. The left part shows the backbone network that encodes user and item features into hidden representations. The middle part illustrates the distribution tokenization and the UniMixer module, which performs token mixing and channel mixing to enable interaction between distribution components. The right part details the Per-Token Hard MOE mechanism used in token mixing.

layer into a context token $\mathbf{z}_0$, which is used to inject sample-specific contextual information into each component token:

$$\mathbf{z}_0 = \text{Proj}(\mathbf{h}) = \text{LayerNorm}(\text{ReLU}(W_c\mathbf{h} + b_c)) \in \mathbb{R}^d \quad (7)$$

The context token and component tokens are concatenated to form the complete token sequence:

$$\mathbf{Z}^{(0)} = [\mathbf{z}_0; \mathbf{z}_1; \ldots; \mathbf{z}_K] \in \mathbb{R}^{(1+K)\times d} \quad (8)$$

Subsequently, the token sequence is fed into the UniMixer module for deep interaction. UniMixer adopts the Mixer architecture [20], [30], achieving information flow and feature enhancement between components through alternating Token Mixing and Channel Mixing operations. For the l-th UniMixer block, the computation process is:

$$\mathbf{Y}^{(l)} = \mathbf{Z}^{(l-1)} + \text{TokenMixing}\left(\text{LayerNorm}(\mathbf{Z}^{(l-1)})\right) \quad (9)$$

$$\mathbf{Z}^{(l)} = \mathbf{Y}^{(l)} + \text{ChannelMixing}\left(\text{LayerNorm}(\mathbf{Y}^{(l)})\right) \quad (10)$$

where TokenMixing adopts a Per-Token Hard MOE mechanism, and ChannelMixing applies independent Channel MLPs to each token. The token mixing operation aggregates information along the token dimension (i.e., the component dimension), enabling each distribution component to perceive the states of other components and contextual information, thereby learning collaborative relationships between components. The channel mixing operation performs feature transformation along the channel dimension, enhancing the representation capability of each token. Compared to self-attention mechanisms, the Mixer architecture has lower computational complexity with respect to the number of components ($O(K)$ vs. $O(K^2)$), making it more suitable for latency-sensitive deployment requirements in industrial recommendation systems [31].

After iterative processing through L UniMixer blocks, we extract the fused component tokens for subsequent parameter generation:

$$\tilde{\mathbf{Z}} = \text{LayerNorm}(\mathbf{Z}^{(L)})_{1:K} \in \mathbb{R}^{K\times d} \quad (11)$$

3) Mixture Parameter Generator: Based on the fused token representations, we generate the parameters of each distribution component through independent prediction heads.

For the exponential distribution component, its rate parameter and mixture weight are generated directly from the hidden representation $\mathbf{h}$:

$$\lambda(\mathbf{x}) = \text{softplus}(W_\lambda\mathbf{h} + b_\lambda) + \epsilon \quad (12)$$

$$\pi_0^{\text{logit}}(\mathbf{x}) = W_{\omega_0}\mathbf{h} + b_{\omega_0} \quad (13)$$

where the softplus activation function $\text{softplus}(z) = \log(1 + e^z)$ ensures the rate parameter is positive, and ϵ is a numerical stability constant (typically 10^{-6}).

For the k-th log-normal component, its parameters are generated from the fused component token $\tilde{\mathbf{z}}_k$:

$$\mu_k(\mathbf{x}) = W_\mu\tilde{\mathbf{z}}_k + b_\mu \quad (14)$$

$$\sigma_k(\mathbf{x}) = \text{softplus}(W_\sigma\tilde{\mathbf{z}}_k + b_\sigma) + \epsilon \quad (15)$$

$$\pi_k^{\text{logit}}(\mathbf{x}) = W_\omega\tilde{\mathbf{z}}_k + b_\omega \quad (16)$$

The log-normal distribution parameters μ_k and σ_k represent the mean and standard deviation of $\log t$ under the normal distribution, respectively. Since $\log t$ can take any real value, μ_k does not require a positivity constraint; while σ_k, as a standard deviation, must be positive, hence the softplus activation is used.

Finally, the mixture weights are obtained through softmax normalization:

$$[\omega_0(\mathbf{x}), \ldots, \omega_K(\mathbf{x})] = \text{softmax}([\pi_0^{\text{logit}}, \ldots, \pi_K^{\text{logit}}]) \quad (17)$$

Combining the above parameters, the complete mixture distribution conditioned on input features is:

$$p(t \mid \mathbf{x}) = \omega_0(\mathbf{x}) f_{\text{exp}}(t \mid \lambda(\mathbf{x})) + \sum_{k=1}^{K} \omega_k(\mathbf{x}) f_{\text{logn}}(t \mid \mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \quad (18)$$

D. Training Objective

We adopt a multi-objective joint optimization strategy to train UniMixer, where each loss function focuses on different aspects of the prediction task.

1) Negative Log-Likelihood Loss: The primary objective is to maximize the likelihood [32] of observed watch times under the mixture distribution, minimizing the negative log-likelihood. For the log-normal component, its log probability density is:

$$\begin{aligned}\log f_{\text{logn}}(t \mid \mu, \sigma^2) = &-\frac{(\log t - \mu)^2}{2\sigma^2} \\ &- \log t - \frac{1}{2}\log(2\pi\sigma^2)\end{aligned} \tag{19}$$

Combined with Eq. (18), the negative log-likelihood loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{NLL}} &= -\frac{1}{N}\sum_{i=1}^{N}\log p(t_i \mid \mathbf{x}_i) \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\left[\sum_{k=0}^{K}\omega_k(\mathbf{x}_i) f_k(t_i)\right]\end{aligned} \tag{20}$$

where $f_0 = f_{\text{exp}}$ and $f_k = f_{\text{logn}}$ (for $k \geq 1$). In implementation, we adopt the log-sum-exp trick to ensure numerical stability.

2) Regression Loss: For effective point estimation, we incorporate a regression objective that minimizes the discrepancy between the predicted distributional mean and observed watch time. The mean of our mixture model takes the form:

$$\begin{aligned}\hat{t}_i &= \mathbb{E}[p(t \mid \mathbf{x}_i)] \\ &= \omega_0(\mathbf{x}_i)\,\frac{1}{\lambda(\mathbf{x}_i)} \\ &\quad + \sum_{k=1}^{K}\omega_k(\mathbf{x}_i)\,\exp\left(\mu_k(\mathbf{x}_i) + \frac{\sigma_k^2(\mathbf{x}_i)}{2}\right)\end{aligned} \tag{21}$$

where the exponential component contributes $1/\lambda$ and each log-normal component contributes $\exp(\mu+\sigma^2/2)$ to the overall expectation. Given the long-tailed nature of watch time data, we apply a logarithmic transformation before computing the loss:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\left|\log(1+\hat{t}_i) - \log(1+t_i)\right| \tag{22}$$

3) Entropy Regularization Loss: To prevent component collapse and maintain expressiveness across all mixture elements, we apply an entropy-based regularizer [33] on the mixing coefficients:

$$\mathcal{L}_{\text{ent}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K}\omega_k(\mathbf{x}_i)\log\omega_k(\mathbf{x}_i) \tag{23}$$

By minimizing this term, we effectively maximize the entropy of the weight allocation, promoting balanced utilization of the available distributional components.

4) Combined Loss Function: Our complete training objective integrates all three terms through weighted summation [19], [34]:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \alpha\mathcal{L}_{\text{reg}} + \beta\mathcal{L}_{\text{ent}} \tag{24}$$

Here, α and β serve as balancing coefficients. The NLL term $\mathcal{L}_{\text{NLL}}$ captures the underlying distributional structure, $\mathcal{L}_{\text{reg}}$ enforces precise mean predictions, and $\mathcal{L}_{\text{ent}}$ preserves diversity among mixture components.

IV. Experiments and Results

In this section, we conduct comprehensive experiments to evaluate the effectiveness of our proposed UniMixer model. We first describe the experimental setup, including datasets, baselines, evaluation metrics, and implementation details. We then present the main results comparing UniMixer with state-of-the-art methods, followed by ablation studies to analyze the contribution of each component.

A. Experimental Setup

1) Datasets: To evaluate the effectiveness of our method, we conduct experiments on two public real-world datasets collected from the recommendation/impression logs of Kuaishou and WeChat Channels, respectively. These datasets have been widely used as benchmarks for watch-behavior and watch-time related tasks. The details of these two datasets are as follows:

- **KuaiRec** [8]. KuaiRec is a real-world dataset collected from the recommendation logs of the short-video platform Kuaishou, featuring relatively comprehensive interaction coverage, each user has watched each video and left feedback. The dataset contains 7,176 users, 10,728 videos, and 12,530,806 impression/interaction records.
- **WeChat**. The WeChat dataset is released by the WeChat Big Data Challenge 2021 and contains impression and interaction logs from WeChat Channels over two weeks. It includes 20,000 users, 96,418 videos, and 7,310,108 interaction records.

For both datasets, we follow the standard preprocessing pipeline and chronologically split the data into training (80%), validation (10%), and test (10%) sets to simulate real-world deployment scenarios.

2) Baselines: We compare our approach with representative state-of-the-art methods for video watch time prediction, including Value Regression (VR), TPM [5], D2Q [13], CREAD [17], and D^{2}CO [14]. Comprehensive descriptions of these baseline models are provided in Section II.

3) Evaluation Metrics: We evaluate our models' prediction performance in terms of both numerical accuracy and ranking capability, the latter of which is particularly important for selecting videos to recommend in practical systems. Following prior work [13], [16], [22], we adopt MAE for regression accuracy and XAUC [13] for ranking consistency.

TABLE I
PERFORMANCE COMPARISON AMONG DIFFERENT APPROACHES ON KUAIREC AND WECHAT DATASET.

Dataset	Method	MAE ↓		XAUC ↑	
		Value	Improv.	Value	Improv.
KuaiRec	VR	4.730	-	0.5379	-
	TPM	5.922	-25.20%	0.5510	2.44%
	D2Q	4.583	3.11%	0.5546	3.10%
	D^2CO	5.087	-7.55%	0.5473	1.75%
	CREAD	4.427	6.41%	0.5821	8.22%
	UniMixer (Ours)	**4.132**	**12.65%**	**0.6110**	**13.59%**
WeChat	VR	38.95	-	0.6009	-
	TPM	21.47	44.88%	0.6055	0.77%
	D2Q	20.36	47.73%	0.6270	4.34%
	D^2CO	20.75	46.73%	0.6257	4.13%
	CREAD	19.21	50.68%	0.6601	9.85%
	UniMixer (Ours)	**18.13**	**53.45%**	**0.6733**	**12.05%**

Best and second best results are in **bold** and underline. ↑: higher is better; ↓: lower is better. Results averaged over 5 runs.

B. Main Results

Our UniMixer model achieves the best performance across all metrics and datasets compared to the current state-of-the-art methods D^2CO and CREAD. Specifically, on the KuaiRec dataset, UniMixer achieves a significant improvement of 4.96% in XAUC and 6.67% reduction in MAE compared to the second-best baseline. On the WeChat dataset, UniMixer improves XAUC by 2.00% and reduces MAE by 5.62%.

These results support our hypothesis that the multi-distribution mixture framework based on exponential-log-normal distributions not only improves overall prediction accuracy but also enhances the model's ability to capture users' relative preferences in a more stable and consistent manner. The substantial improvements in XAUC demonstrate that UniMixer can better preserve the ranking order of user preferences, which is crucial for recommendation systems where the goal is to select the most engaging videos for users.

C. Ablation Study

To validate the contribution of each component, we conduct ablation studies on KuaiRec by removing or modifying key elements: (1) Model Architecture: removing channel mixing or the entire Mixer module; (2) Distribution Design: removing the exponential component or using only exponential; (3) Loss Function: disabling regression loss ($\alpha = 0$) or entropy regularization ($\beta = 0$).

The ablation results in Table II reveal several important findings:

Importance of UniMixer Module. Removing the channel mixing operation causes the most significant performance degradation among all architectural variants (MAE increases by 11.19%, XAUC drops from 0.6110 to 0.5635), demonstrating that cross-component interaction is critical for capturing the complex dependencies between distribution components.

TABLE II
ABLATION STUDY RESULTS OF UNIMIXER ON KUAIREC DATASET

Ablated Method	MAE		XAUC	
	value	diff	value	diff
UniMixer	4.1318	-	0.6110	-
Model Architecture:				
$\ominus$ Channel Mixing	4.5940	+11.19%	0.5635	-7.77%
$\ominus$ Mixer	4.4916	+8.71%	0.5704	-6.64%
Distribution Design:				
$\ominus$ Exp	4.3773	+5.94%	0.5821	-4.73%
Exp Only	4.5051	+9.03%	0.5715	-6.46%
Loss Function:				
$\alpha = 0$	4.3890	+6.23%	0.5744	-5.99%
$\beta = 0$	4.4018	+6.54%	0.5781	-5.39%

Removing the entire Mixer module also leads to substantial degradation (MAE +8.71%, XAUC -6.64%), validating the effectiveness of our proposed fusion mechanism.

Effectiveness of Mixture Distribution Design. The comparison between "UniMixer", "$\ominus$ Exp", and "Exp Only" variants reveals that the combination of exponential and log-normal components is beneficial. Using only the exponential component significantly hurts both metrics (MAE +9.03%, XAUC -6.46%), as it cannot capture the diverse long-tail watching patterns. Removing the exponential component causes relatively smaller degradation (MAE +5.94%, XAUC -4.73%), suggesting that while the exponential component contributes to modeling quick-skip behaviors, the log-normal mixture plays a more fundamental role in capturing the overall watch time distribution.

Role of Loss Components. Setting $\alpha = 0$ (removing regression loss) leads to notable performance drops (MAE +6.23%, XAUC -5.99%), indicating that the regression loss helps regularize the distribution expectation and improves prediction accuracy. Setting $\beta = 0$ (removing entropy regularization) causes similar degradation (MAE +6.54%, XAUC -5.39%), suggesting that without entropy regularization, the model may suffer from mode collapse where it over-relies on fewer mixture components. This validates the importance of both loss components for maintaining prediction quality and component diversity.

V. CONCLUSION

In this paper, we addressed the challenging problem of watch time prediction in short-video recommendation systems by identifying two fundamental challenges: the unobservability of latent multi-duration distributions and the inter-distribution cross-dependency arising from the complex interplay between user preferences and video content structures. To tackle these challenges, we proposed UniMixer, a novel framework that models watch time as a random variable following an Exponential-Log-Normal mixture (ELN) distribution, where

each distribution component is parameterized as a learnable token and deeply interacts with contextual features through a unified token-mixing mechanism. This design enables explicit capture of high-order dependencies among different viewing patterns while maintaining linear computational complexity suitable for real-time industrial deployment. Extensive experiments on two large-scale public datasets (KuaiRec and WeChat) demonstrated that UniMixer achieves state-of-the-art performance on both ranking (XAUC) and regression (MAE) metrics, with ablation studies validating the critical role of token mixing and the effectiveness of the mixture distribution design.

VI. Acknowledgements

The AI assistant, Gemini 3, is used solely for refining the writing of our paper.

References

[1] Q. Cai, S. Liu, X. Wang, T. Zuo, W. Xie, B. Yang, D. Zheng, P. Jiang, and K. Gai, "Reinforcing user retention in a billion scale short video recommender system," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 421–426.

[2] Y. Pan, C. Gao, J. Chang, Y. Niu, Y. Song, K. Gai, D. Jin, and Y. Li, "Understanding and modeling passive-negative feedback for short-video sequential recommendation," in *Proceedings of the 17th ACM conference on recommender systems*, 2023, pp. 540–550.

[3] M. Savic, "Research perspectives on tiktok & its legacy apps— from musical. ly to tiktok: Social construction of 2020's most downloaded short-video app," *International Journal of Communication*, vol. 15, p. 22, 2021.

[4] Y.-H. Wang, T.-J. Gu, and S.-Y. Wang, "Causes and characteristics of short video platform internet community taking the tiktok short video application as an example," in *2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. IEEE, 2019, pp. 1–2.

[5] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[6] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The youtube video recommendation system," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010, p. 293–296.

[7] Y. Liu, Q. Liu, Y. Tian, C. Wang, Y. Niu, Y. Song, and C. Li, "Concept-aware denoising graph neural network for micro-video recommendation," in *Proceedings of the 30th ACM International Conference on Information Knowledge Management*, 2021, p. 1099–1108.

[8] C. Gao, S. Li, W. Lei, J. Chen, B. Li, P. Jiang, X. He, J. Mao, and T.-S. Chua, "Kuairec: A fully-observed dataset and insights for evaluating recommender systems," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, p. 540–550.

[9] X. Gong, Q. Feng, Y. Zhang, J. Qin, W. Ding, B. Li, P. Jiang, and K. Gai, "Real-time short video recommendation on mobile devices," in *Proceedings of the 31st ACM international conference on information & knowledge management*, 2022, pp. 3103–3112.

[10] T. Wang, J. Chen, F. Zhuang, L. Lin, F. Xia, L. Du, and Q. He, "Capturing attraction distribution: Sequential attentive network for dwell time prediction," in *ECAI 2020*. IOS Press, 2020, pp. 529–536.

[11] S. Wu, M.-A. Rizoiu, and L. Xie, "Beyond views: Measuring and predicting engagement in online videos," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, 2018.

[12] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan, "Beyond clicks: dwell time for personalization," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 113–120.

[13] R. Zhan, C. Pei, Q. Su, J. Wen, X. Wang, G. Mu, D. Zheng, P. Jiang, and K. Gai, "Deconfounding duration bias in watch-time prediction for video recommendation," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 4472–4481.

[14] H. Zhao, L. Zhang, J. Xu, G. Cai, Z. Dong, and J.-R. Wen, "Uncovering user interest from biased and noised watch time in video recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 528–539.

[15] H. Zhao, G. Cai, J. Zhu, Z. Dong, J. Xu, and J.-R. Wen, "Counteracting duration bias in video recommendation via counterfactual watch time," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4455–4466.

[16] X. Lin, X. Chen, L. Song, J. Liu, B. Li, and P. Jiang, "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*, 2023, pp. 4497–4506.

[17] J. Sun, Z. Ding, X. Chen, Q. Chen, Y. Wang, K. Zhan, and B. Wang, "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, no. 8, 2024, pp. 9027–9034.

[18] S. Yang, H. Yang, L. Du, A. Ganesh, B. Peng, B. Liu, S. Li, and J. Liu, "Swat: Statistical modeling of video watch time through user behavior analysis," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, 2025, p. 2768–2778.

[19] X. Zhao, R. Ma, J. Chen, W. Zhao, P. Yang, and Y. Hu, "Multi-granularity distribution modeling for video watch time prediction via exponential-gaussian mixture network," in *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, 2025, p. 309–318.

[20] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2023, pp. 459–469.

[21] X. Wang, T. Liu, and J. Miao, "A deep probabilistic model for customer lifetime value prediction," *arXiv preprint arXiv:1912.07753*, 2019.

[22] C. Lin, S. Liu, C. Wang, and Y. Liu, "Conditional quantile estimation for uncertain watch time in short-video recommendation," *arXiv preprint arXiv:2407.12223*, 2024.

[23] C. Lin, C. Wang, A. Xie, W. Wang, Z. Zhang, C. Ruan, Y. Huang, and Y. Liu, "Alignpxtr: Aligning predicted behavior distributions for bias-free video recommendations," *arXiv preprint arXiv:2503.06920*, 2025.

[24] H. Ma, K. Tian, T. Zhang, X. Zhang, C. Chen, H. Li, J. Guan, and S. Zhou, "Sequence generation modeling for continuous value prediction," in *WWW*, 2025.

[25] K. Balakrishnan, *Exponential distribution: theory, methods and applications*. Routledge, 2019.

[26] E. L. Crow and K. Shimizu, *Lognormal distributions*. Marcel Dekker New York, 1987.

[27] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.

[28] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] J. Zhu, Z. Fan, X. Zhu, Y. Jiang, H. Wang, X. Han, H. Ding, X. Wang, W. Zhao, Z. Gong, H. Yang, Z. Chai, Z. Chen, Y. Zheng, Q. Chen, F. Zhang, X. Zhou, P. Xu, X. Yang, D. Wu, and Z. Liu, "Rankmixer: Scaling up ranking models in industrial recommenders," in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025, p. 6309–6316.

[31] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[32] J.-X. Pan, K.-T. Fang, J.-X. Pan, and K.-T. Fang, "Maximum likelihood estimation," *Growth curve models and statistical diagnostics*, pp. 77–158, 2002.

[33] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," *Advances in neural information processing systems*, vol. 33, pp. 16 096–16 107, 2020.

[34] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite mixture models," *Annual review of statistics and its application*, vol. 6, no. 1, pp. 355–378, 2019.