# Learning Fine-grained User Interests for Micro-video Recommendation

Yu Shang
Department of Electronic Engineering, Tsinghua University
Beijing, China
shangy21@mails.tsinghua.edu.cn

Chen Gao
Department of Electronic Engineering, Tsinghua University
Beijing, China
chgao96@gmail.com

Jiansheng Chen*
University of Science and Technology Beijing
Beijing, China
jschen@ustb.edu.cn

Depeng Jin
Department of Electronic Engineering, Tsinghua University
Beijing, China
jindp@tsinghua.edu.cn

Meng Wang
School of Computer Science and Information Engineering,Hefei University of Technology
Hefei, China
eric.mengwang@gmail.com

Yong Li
Department of Electronic Engineering, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

## ABSTRACT

Recent years have witnessed the rapid development of online micro-video platforms, in which the recommender system plays an essential role in overcoming the information overloading problem and providing personalized content for users. Although some progress has been achieved in the micro-video recommendation, there are still some limitations in learning the representations of user interests and video features. Specifically, the user modeling in existing works is performed at a coarse-grained level, *i.e.*, video level. However, in micro-video recommendation, the user feedback is at a continuous form—users can skip over a video at each frame—which reveals fine-grained user preferences. In this work, we approach the problem of learning fine-grained user preferences for micro-video recommendation by first collecting two real-world datasets. To address the challenges of preference modeling and weak supervision signal, we propose a solution named FRAME (short for **F**ine-g**RA**ined preference-modeling for **M**icro-video r**E**commendation). Specifically, we first adopt visual feature extraction and transformation to maintain the fine-grained video embeddings. We then propose graph convolution layers to learn the user preference from complex and fine-grained user-clip relations, and hybrid-supervision objectives for enhancing the supervision signal. The experimental results on two collected real-world datasets demonstrate the effectiveness of our proposed model. We release the datasets and codes in https://github.com/tsinghua-fib-lab/FRAME, which we believe can benefit the community.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**;

*The corresponding authors.

This work is licensed under a Creative Commons Attribution International 4.0 License.

*SIGIR '23, July 23–27, 2023, Taipei, Taiwan*
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591713

## KEYWORDS

Micro-video Recommendation; Fine-grained User Interest Modeling; Fine-grained Video Features

**ACM Reference Format:**
Yu Shang, Chen Gao, Jiansheng Chen, Depeng Jin, Meng Wang, and Yong Li. 2023. Learning Fine-grained User Interests for Micro-video Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591713

## 1 INTRODUCTION

Micro-video applications such as TikTok are increasingly popular nowadays, with billions of active users each day. Its great success largely owns to recommender system, which learns user preference from behavioral data and effectively filters videos. Existing works of traditional recommender systems are limited to the explicit feedback such as rating or implicit feedback such as click, purchase, etc. However, the feedback in micro-video recommendation is brand new, which is presented in a continuous form. Precisely, the users' browsing behavior can end at any specific position/frame of a video (when users choose to skip over), yielding a special preference signal. Therefore, it is not reasonable to directly consider the whole video as a positive or negative one.

Although micro-video recommendation is a widely-explored topic [4, 23, 26, 27, 31, 42], existing works only introduce the video-level preferences, such as [42] which extracts content features from the whole video or [23, 27] which uses the embedding of thumbnail as the video feature. That is, existing user-video interaction only contains whether the user skips or does not skip a video, but does not consider which clip the user skips, which is coarse-grained. While the modeling of fine-grained user-video feedback, which is critical for micro-video recommendation, is still less-explored. Different from traditional user feedback, fine-grained feedback considers the skipping behavior together with corresponding fine-grained video content, which could benefit modeling user preference on a more fine-grained level (e.g., clip-level). As a brand new kind of feedback, learning fine-grained user interests for micro-video recommendation suffers from the following challenges.

- **Fine-grained feature learning.** Learning fine-grained user preferences is supported by fine-grained video features, such as clip-level or even frame-level. One reason for the missing literature is that there is no public dataset consisting of both fine-grained user feedback and video features.
- **The user-video relations are complex.** Since the user feedback towards the video is fine-grained, there exist more complex user-video relations. When a user interacts with a video, he/she may have only watched a fraction of the video, while the remaining is not involved. It is challenging to learn from the new kind of user-video relation.
- **The supervision signal is weak.** The continuous feedback, different from explicit feedback or implicit feedback in existing recommenders, reveals very weak prediction signals. A video that the user skips over indeed contains two parts of content that matches and does not match user preferences at the same time.

To address the challenges mentioned above, in this work, we collect two real-world large-scale datasets from a mainstream micro-video platform. The datasets differ from existing public datasets on the users' fine-grained feedback including the user's watching time of interacted videos and fine-grained visual pixel data of videos. Aiming to learn fine-grained user and video representation, we then propose a graph neural network-based solution named FRAME (short for **F**ine-g**RA**ined preference-modeling for **M**icro-video r**E**commendation). Specifically, we first obtain the fine-grained video features via pre-trained neural networks, based on which we deploy a simple yet effective transformation layer to obtain fine-grained video embeddings. We then construct a user-clip relation graph, which contains two kinds of user fine-grained feedback: skip and non-skip, reflecting weak-negative and weak-positive user preferences, respectively. Then the graph convolutional layers on these two kinds of relations are deployed to learn user representations, which are further merged for the prediction of user-video interaction. We finally propose hybrid objectives for enhancing the supervision signals. The contribution of our work can be summarized as follows:

- We take the first step to propose a new paradigm of fine-grained feedback learning, which is seldom considered in the existing literature of recommender systems. We carefully collect two real-world large-scale datasets with fine-grained user-video interaction and video features and release the datasets.
- We propose a novel solution named FRAME for the new problem. We first combine video feature extraction into preference learning and then propose a graph convolutional network-based approach to extract user fine-grained preferences. Lastly, we introduce a hybrid loss into the model training, which captures the user decision process well when interacting with micro-videos.
- Extensive experiments on two real-world datasets verify the effectiveness of our proposed method. Further experiments of ablation study demonstrate the rationality of our method's each component. We have also conducted case studies with insightful conclusions about how our proposed model captures user preference in a fine-grained manner.

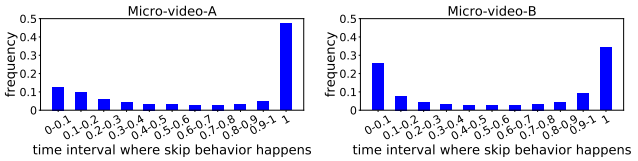



**Figure 1: The distribution of the skip-behavior time-interval.**

## 2 PROBLEM MOTIVATION AND DEFINITION

Compared to traditional interactions like click behavior in other recommender systems, users' interaction manners in micro-video recommendation differ a lot. Specifically, a user can skip a video at any time, or does not take any action (then the next video will be played). These behaviors reflect the user interests and preferences in such a continuous video-playing process.

First, we conduct some data analysis on a real-world micro-video recommendation dataset[1]. We present the distribution of the skip behavior's interval in Figure 1. For example, if the video length is 10 seconds, "0.2-0.3" means the user's skip behavior happens at the interval of 2-3 seconds. We can observe from the distribution that the number of skip behaviors is more than non-skip behaviors. Besides, the users may skip the video at any time interval. Such data characteristics require a new paradigm of modeling user-video interactions to capture the fine-grained signal expressed by users' continuous feedback. Motivated by the findings of data analysis, we define the problem of learning fine-grained user preferences for micro-video recommendation.

Let $\mathcal{U} = \{u_1, u_2, \cdots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ denote the user set and the video set, respectively, in which $M$ and $N$ represent the number of users in $\mathcal{U}$ and videos in $\mathcal{V}$. We format each user's historical interaction as a set of triplets, $\mathcal{I}^{(u)} = \{(u, v_i, r_i)_{i=1}^{m}\}$ where $r_i$ represents the ratio of the user $u$'s watching time to the duration time of the video $v_i$. For example, $(u_1, v_1, 0.6)$ means that user $u_1$ watched the first 60% of video $v_1$ and $(u_2, v_2, 1)$ means user $u_2$ watched all of video $v_2$. Besides, the fine-grained user-preference modeling relies on the fine-grained video features (clip-level or even frame-level), which means the videos' raw features are required.

The studied problem can be formulated as follows[2]:

**Input:** the interaction data $\mathcal{I}^{(u)}$ of each user $u \in \mathcal{U}$; videos' raw features X *i.e.*, image pixels of frames.

**Output:** A micro-video recommendation model that estimates the probability that the user $u$ with interaction set $\mathcal{I}^{(u)}$ will like the given video $v$, of which "like" follows a widely-accepted definition of watching 100% of the given video.

## 3 METHODOLOGY

Our proposed FRAME (shown in Figure 2) has four components.

- **Visual-enhanced embedding layer.** To model the fine-grained user interest and video features, we first extract visual features of clips in each video by a convolutional neural network rather than the thumbnail widely used in existing works of micro-video recommendation [23, 27, 42]. In addition, since not all vision

[1]The dataset details will be introduced in experiments (Section 4).
[2]For better understanding, notations used in this paper are summarized in Table 1.

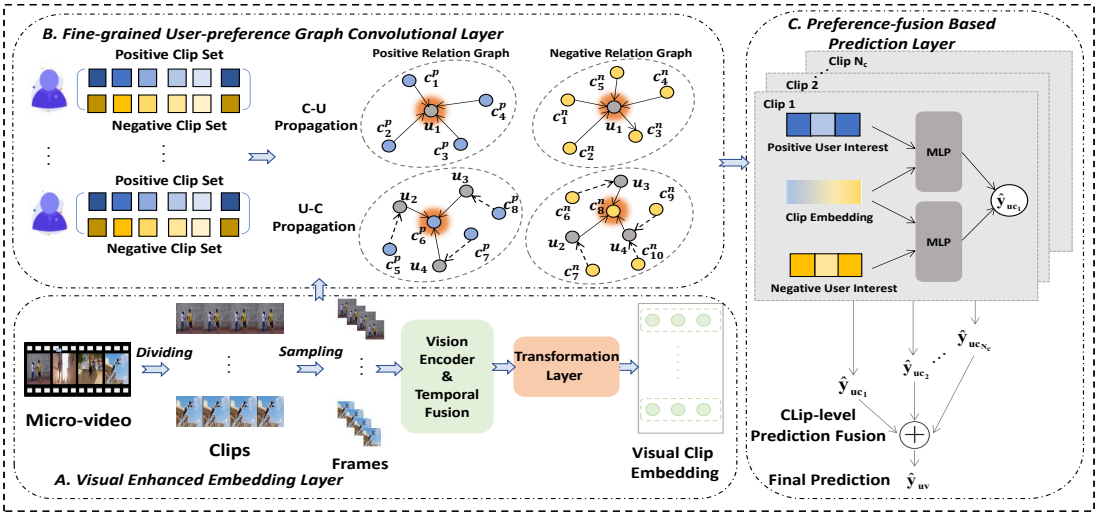



**Figure 2: Illustration of the FRAME model. Clip-level visual features are extracted and enhanced (A) to construct the fine-grained micro-video representation. Based on the fine-grained video features, a graph convolutional layer on the user-clip relation is applied to aggregate users' positive and negative interests (B). Finally the video-level prediction is aggregated from the clips to the video, where we adaptively fuse positive and negative preferences for the final prediction (C).**

**Table 1: The description of commonly used notations.**

| Notations | Description |
|---|---|
| $\mathcal{U}, \mathcal{V}, \mathcal{I}$ | The set of users, videos and interactions. |
| $M, N, C$ | The number of total users, videos and clips. |
| $\mathbf{f}_c, \mathbf{e}_c$ | The original and enhanced visual feature of clips. |
| $C_u^p, C_u^n$ | The positive and negative clip set of user $u$. |
| $N_+^{(u)}, N_-^{(u)}$ | The number of videos fully watched and skipped by user $u$. |
| $\mathbf{R}^p, \mathbf{R}^n$ | The positive and negative user-clip interaction matrix. |
| $\tilde{\mathbf{R}}^p, \tilde{\mathbf{R}}^n$ | Normalized positive and negative user-clip interaction matrix. |
| $\mathbf{H}_c$ | The clip embedding matrix for prediction. |
| $\mathbf{H}_u^p, \mathbf{H}_u^n$ | Final positive and negative user interest embedding matrix. |
| $\alpha_p, \alpha_n$ | The weight of positive and negative clip-level result. |
| $\alpha, \beta$ | The weight of point-wise and pair-wise loss. |

features will affect users' behaviors, we deploy a simple yet effective transformation operation to obtain the fine-grained video embeddings, *i.e.* the embeddings of video clips.

- **Fine-grained user-preference graph convolutional layer.** Based on the fine-grained video features, we can further conduct fine-grained user-preference learning. As the videos are split into clips, we can build the user-clip relation, of which there are two kinds, skip (negative) and non-skip (positive). We then propose a graph convolutional layer on the user-clip relation to aggregate users' positive and negative interests.
- **Preference-fusion based prediction Layer.** Standing on the shoulder of the user's positive and negative preference and enhanced clip features of each micro-video, we make clip-level predictions via a multi-layer perception (MLP) network, respectively. The video-level prediction is aggregated from the clip to the video. Here we adaptively fuse positive and negative preferences for the final prediction.
- **Hybrid supervision learning.** To train our model, we design two different supervision learning methods. On the one hand, we adopt an improved binary cross-entropy loss to supervise user clip-level interest. On the other hand, we propose utilizing the pair-wise BPR loss [34] on the fine-grained preferences on the different clips of the same video.

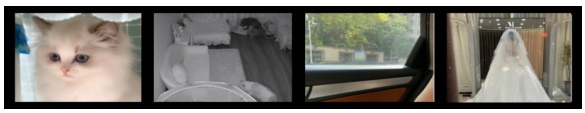


**Figure 3: A real example of the content difference between different clips in the same video.**

## 3.1 Visual-enhanced Embedding Layer

The visual content of micro-videos plays a leading role in their multimodal information. Thus most existing works of micro-video recommendation [4, 23, 27] use visual features of thumbnail to represent the video content. However, even though for one video, its content varies between different clips as revealed by the example in Figure 3. Therefore, it is essential to explore extracting fine-grained features of videos and adopt them into the recommendation model.

*3.1.1 Visual Feature Extraction.* Aiming to model the fine-grained user interest and video features, we use a pre-trained 2D-CNN architecture ResNet-50 [18] as the vision encoder $\mathcal{F}_v$, similar to existing works [27, 42]. Specifically, we reserve the first five CNN layers of the pre-trained ResNet-50 and add a pooling layer to get the feature vectors. To reduce the dimension of the visual feature, a fully connected layer is further deployed in the end. For each clip

$c$, we uniformly sample $K$ frames and get $K$ frame-level visual features. A fusion layer (here we use mean-pooling) is made use of to obtain the visual feature of the clip by aggregating its frames' visual feature as follows,

$$\mathbf{f}_c = \textbf{Aggregate}\left(\mathcal{F}_v(X_{f_i})|f_i \in c, i \in (1, K)\right), \tag{1}$$

where $f_i$ denotes the $i$-th frame in the clip $c$ and $X_{f_i} \in \mathbb{R}^{h\times w\times d_c}$ ($d_c$ is the number of channels) represents the image of the $i$-th frame. We take the obtained clip visual embedding $\mathbf{f}_c \in \mathbb{R}^d$ as the initial clip embedding in our recommendation model.

*3.1.2 Visual Feature-enhanced Embedding Layer.* Considering in recommender systems, not all vision features such as the background scene of one frame will affect user behaviors. Therefore, we feed the clips' visual features into a transformation layer to obtain the embeddings in the user-preference space. It can help reserve useful signals for preference learning. Specifically, we make a transformation with the embedding dimension unchanged as $\mathbf{e}_c = \mathbf{W}_t\mathbf{f}_c$, where $\mathbf{f}_c \in \mathbb{R}^d$ and $\mathbf{e}_c \in \mathbb{R}^d$ are the original visual embedding and enhanced embedding of the clip $c$ respectively, and $\mathbf{W}_t \in \mathbb{R}^{d\times d}$ is a learnable transformation matrix to map the visual embedding to an enhanced form.

## 3.2 Fine-grained User-preference Graph Convolutional Layer

With the first challenge addressed by the above embedding layer, we then propose a graph convolutional layer to learn from the complex user-video relationships. Different from traditional user-item interactions which mainly include explicit and implicit feedback, in micro-video recommendation, the user-video relations are in a continuous and fine-grained form.

*3.2.1 Positive and Negative interest set Construction.* Although user-video relation is complex, and we even cannot determine whether an interaction is positive or negative, luckily the user-clip relations are easier to handle. We use $N_+^{(u)}$ and $N_-^{(u)}$ to represent the number of videos fully watched and skipped by the user $u$, respectively. Meanwhile let $N_c$ denotes the number of clips in each video, we collect all clips from the fully watched videos as the positive clip set of the user $u$, which is formulated as:

$$C_u^p = \bigcup_{i=1}^{N_+^{(u)}} \{c_{i,1}^p, c_{i,2}^p, \cdots, c_{i,N_c}^p\}, \tag{2}$$

As for the negative set, we take the clips during which the skipping behavior happens as the negative clips for users. The reason is that users' interests may drop to some extent when they are watching a certain clip, at which moment they tend to skip the video. Then we generate the negative clip set of the user $u$ as follows:

$$C_u^n = \{c_1^n, c_2^n, \cdots, c_{N_-^{(u)}}^n\}, \tag{3}$$

where $c_1^n$ is the clip from the video $v_1^n$ in which the skipping behavior of user $u$ happens and other clips are the same.

*3.2.2 User-Clip Graph Construction.* As mentioned above, we have constructed both positive and negative user-clip interaction for each user, which can be re-constructed to a user-clip interaction graph.

Let $C$ denotes the number of total clips, then the interaction matrix $\mathbf{R}^p(\mathbf{R}^n) \in \mathbb{R}^{M\times C}$ is applied to describe the positive (negative) relationship between users and clips, where $M$ is the number of users as mentioned in previous sections. The elements of the matrix are defined as follows:

$$R_{ij}^p = \begin{cases} 1, & c_j \in C_{u_i}^p; \\ 0, & \text{otherwise}; \end{cases} \quad R_{ij}^n = \begin{cases} 1, & c_j \in C_{u_i}^n; \\ 0, & \text{otherwise}; \end{cases} \tag{4}$$

where $C_{u_i}^p$ and $C_{u_i}^n$ are the positive and negative clip set of the user $u_i$. We then obtain the dual-side adjacency matrix of the user-clip graph as follows,

$$\mathbf{A}^p = \begin{pmatrix} \mathbf{0} & \mathbf{R}^p \\ (\mathbf{R}^p)^\top & \mathbf{0} \end{pmatrix}, \quad \mathbf{A}^n = \begin{pmatrix} \mathbf{0} & \mathbf{R}^n \\ (\mathbf{R}^n)^T & \mathbf{0} \end{pmatrix}. \tag{5}$$

We use $\tilde{A}$ to denote the normalized form of adjacency matrix, and the positive and negative adjacency matrix are formulated as follows respectively:

$$\begin{aligned} \tilde{\mathbf{A}}^p &= (\mathbf{D}^p)^{-\frac{1}{2}}\mathbf{A}^p(\mathbf{D}^p)^{\frac{1}{2}} = \begin{pmatrix} \mathbf{0} & \tilde{\mathbf{R}}^p \\ \left(\tilde{\mathbf{R}}^p\right)^\top & \mathbf{0} \end{pmatrix}, \\ \tilde{\mathbf{A}}^n &= (\mathbf{D}^n)^{-\frac{1}{2}}\mathbf{A}^n(\mathbf{D}^n)^{\frac{1}{2}} = \begin{pmatrix} \mathbf{0} & \tilde{\mathbf{R}}^n \\ \left(\tilde{\mathbf{R}}^n\right)^\top & \mathbf{0} \end{pmatrix}, \end{aligned} \tag{6}$$

where $\mathbf{D}^p$ $(\mathbf{D}^n) \in \mathbb{R}^{(M+C)\times(M+C)}$is a diagonal matrix (the degree matrix), in which each entry $D_{ii}^p$ $(D_{ii}^n)$ denotes the number of nonzero entries in the i-th row of the corresponding adjacency matrix.

*3.2.3 Dual-side user-clip modeling GCN layer.* Inspired by the success of GCN [25] in modeling different-order interactions between nodes by aggregating information from different-hop neighbors [14, 20, 28, 39, 41, 42]. Since user behaviors may be extremely sparse, we do not explicitly assign the user embedding matrix. Instead, we obtain user embedding by the mechanism of embedding propagation in GCN models, through which user embeddings are calculated by aggregating neighbours' embeddings. Specifically, we can obtain users' two parts of embeddings, reflecting positive and negative interests respectively as follows,

$$\mathbf{H}_u^p = \sigma\left(\tilde{\mathbf{R}}^p\mathbf{H}_c^{(0)}\mathbf{W}_p^{(1)}\right), \mathbf{H}_u^n = \sigma\left(\tilde{\mathbf{R}}^n\mathbf{H}_c^{(0)}\mathbf{W}_n^{(1)}\right), \tag{7}$$

where $\mathbf{H}_c^{(0)} \in \mathbb{R}^{C\times d}$ is the clip embedding matrix, which is set as the visual-enhanced clip embedding, $\tilde{\mathbf{R}}^p$ and $\tilde{\mathbf{R}}^n$ are shown in Equation (6) , $\sigma(\cdot)$ denotes the nonlinear activation function. Here $\mathbf{W}_p^{(1)}$ and $\mathbf{W}_n^{(1)} \in \mathbb{R}^{d\times d}$ denote the trainable weight matrix. Thus, we obtain $\mathbf{H}_u^p$ and $\mathbf{H}_u^n$ as the positive and negative user interest embedding matrix, which will be used for prediction later.

In the real-world scenario, digging out the clip-clip relationship could further help improve the effect of clip modeling. For example, if two clips are connected by the same user in the positive or negative interaction graph, they might be close. To capture such higher-order information transmission between clips, we conduct a two-hop embedding propagation, which can be formulated as follows,

$$\mathbf{H}_c^{p(1)} = \sigma\left(\left(\tilde{\mathbf{R}}^p\right)^\top\mathbf{H}_u^p\mathbf{W}_p^{(2)}\right), \mathbf{H}_c^{n(1)} = \sigma\left(\left(\tilde{\mathbf{R}}^n\right)^\top\mathbf{H}_u^n\mathbf{W}_n^{(2)}\right), \tag{8}$$

where $\mathbf{W}_p^{(2)}$ and $\mathbf{W}_n^{(2)} \in \mathbb{R}^{d\times d}$ represent the trainable weight matrix in the second GCN layer. Since positive and negative are defined towards user preferences, actually, here we do not need to distinguish positive and negative for clips. Therefore, we apply a mean pooling to get the final representation of clips as follows,

$$\mathbf{H}_c^{(1)} = \mathbf{Mean}\left(\mathbf{H}_c^{p(1)}, \mathbf{H}_c^{n(1)}\right). \tag{9}$$

To sum up, we have acquired $\mathbf{H}_c^{(0)}$ and $\mathbf{H}_c^{(1)}$ as the embedding of clips with different information and they can work together to represent the clips.

## 3.3 Preference-fusion Based Prediction Layer

After obtaining users' dual-side interest embedding and clips' embedding from different GCN layers, we can now make the prediction. Specifically, we adaptively combine two layers of clip embeddings by the coefficient $\alpha_0$ and $\alpha_1$ to get the final clip embedding used for prediction, which is formulated as follows,

$$\mathbf{H}_c = \alpha_0\mathbf{H}_c^{(0)} + \alpha_1\mathbf{H}_c^{(1)}. \tag{10}$$

Given the clip embedding $\mathbf{h}_c$ from the clip embedding matrix $\mathbf{H}_c$ and the dual-side user interest embedding $\mathbf{h}_u^p$ and $\mathbf{h}_u^n$ from $\mathbf{H}_u^p$ and $\mathbf{H}_u^n$ respectively, first we concatenate the clip embedding with the positive and negative interest separately and then feed them into a multi-layer perception (MLP) network respectively to get the prediction:

$$\hat{y}_{uc}^p = \mathbf{MLP}^p(h_u^p \parallel h_c), \hat{y}_{uc}^n = \mathbf{MLP}^n(h_u^n \parallel h_c). \tag{11}$$

Next we combine both positive and negative results with weight as the prediction result towards the user $u$ and the clip $c$. Different from existing work [27] which gives positive weight to both the dual-side result, we argue that the negative feedback should have a negative weight for fusion. The final clip-level prediction is as follows:

$$\hat{y}_{uc} = \alpha_p\hat{y}_{uc}^p - \alpha_n\hat{y}_{uc}^n. \tag{12}$$

Finally the fusion of each clip's result is set as the probability that the user $u$ might finish watching the given video $v$. Although many choices could be applied, here we conduct a simple yet effective average-pooling operation to obtain the final prediction as follows,

$$\hat{y}_{uv} = \frac{1}{N_c}\sum_{i=1}^{N_c}\hat{y}_{uc_i}, \tag{13}$$

where $N_c$ is the number of clips used to divide the video.

## 3.4 Hybrid Supervision Learning

The target of industrial micro-video recommendation is always defined to maximize the probability that the user will fully watch the video. To address the third challenge of limited supervision signal, we introduce a hybrid loss to enhance the learning procedure.

**User-clip point-wise loss.** First we design an improved point-wise loss to learn user preference from the fine-grained user-clip interactions. Here we choose to optimize the prediction of user-clip score rather than user-video score as the former one carries fine-grained preference information. We regard the clips that users have watched as positive samples and clips ***at which*** users skipped as negative samples. For example, if a user has watched three 10-second clips and then chooses to skip during the fourth 10-second clip, our method considers the first three as weak-positive and the fourth as weak-negative. As for the clips after the skipping behavior's timestamp, we choose to give up endowing labels since we don't know users' attitudes towards these unexposed parts.

It is worth considering that clips in different positions may have different impacts on user interest modeling, but directly modeling the relation will largely increase the computation complexity. Therefore, we try to encode the temporal (position) relation between clips indirectly through the supervision signal design. To be specific, we set different penalty coefficients for clips at different positions to distinguish the strength of these positive samples. Specifically, the penalty coefficient of the i-th clip is set as $\epsilon_i = \frac{i}{N_c}$, which suggests that with the extension of users' watching time, the clip could get higher confidence when it is defined as a positive sample. Taking an example where $N_c = 4$, the penalty coefficient of the first clip in videos (denoted as $\epsilon_1$) is set as 0.25, and the coefficient of other clips is calculated in a similar way. Eventually the improved point-wise loss is formulated as follows:

$$\mathcal{L}_1 = -\sum_{i\in\mathcal{U}}\left(\sum_{j=1}^{N_c}\sum_{k\in C_i^{p,j}}\epsilon_j\log\sigma(\hat{y}_{ik}) + \sum_{k\in C_i^n}\log\left(1-\sigma(\hat{y}_{ik})\right)\right), \tag{14}$$

where $C_i^{p,j}$ is the positive clip set of the user $i$ in which clips are all the j-th clip in one video, $C_i^n$ is the negative clip set of the user $i$.

**Clip-clip pair-wise loss.** To further enhance the supervision signal, we introduce a pair-wise loss. Specifically, in the same video, the clips at which user skipped always reflect more negative preference than the part that user has watched. In other words, the watched part, which may consist of several clips, can be merged as a positive signal taking a mean-pooling operation towards these clips' embedding. Here we employ Bayesian Personalized Ranking (BPR) loss [34] as the pair-wise loss for our training to encourage our model to distinguish the positive and negative clips for the same video. Therefore for each triplet $(u, v, r)$ with $r < 1$, there's a pair represented as $(c_{p,v}^{(u)}, c_{n,v}^{(u)})$. Then the pair-wise loss can be formulated as follows,

$$\mathcal{L}_2 = -\sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{V}_n^{(i)}}\log\sigma\left(\hat{y}_{ic_{p,j}^{(i)}} - \hat{y}_{ic_{n,j}^{(i)}}\right), \tag{15}$$

where $\mathcal{V}_n^{(i)}$ represents the videos skipped by the user $i$.

Finally, we use the joint loss obtained by combining the two types of loss mentioned above as our final objective function, which can be formulated as follows,

$$\mathcal{L} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2 + \lambda\|\Theta\|_2, \tag{16}$$

where $\alpha$ and $\beta$ control the weight of the point-wise loss and pair-wise loss, $\Theta$ is the set of parameters to be regularized, and $\lambda$ is the $L_2$ regularization factor.

## 3.5 Complexity Analysis

We conduct some complexity analysis of our proposed models including memory and time complexity.

*3.5.1 Memory Complexity.* The main memory consumption of our model comes from node embeddings, user-clip adjacency matrix and trainable weights of GCN layers. The node embeddings includes clip embedding $\mathbf{H}_c^{(0)}$ and $\mathbf{H}_c^{(1)} \in \mathbb{R}^{C\times d}$, and positive and negative user interest embedding $\mathbf{H}_u^p$ and $\mathbf{H}_u^n \in \mathbb{R}^{M\times d}$, where

$d$ is the embedding dimension, $M$ and $C$ denotes the number of users and clips, respectively. The dual-side adjacency matrix $\tilde{\mathbf{A}}^p$ and $\tilde{\mathbf{A}}^n \in \mathbb{R}^{(M+C)\times(M+C)}$, we use sparse matrix to store the matrix, extremely decreasing the memory consumption. By comparison, the trainable parameters of GCN layers $\mathbf{W}_p^{(1)}$, $\mathbf{W}_n^{(1)}$, $\mathbf{W}_p^{(2)}$ $\mathbf{W}_n^{(2)} \in \mathbb{R}^{d\times d}$ consumes much less memory. To sum up, it will not bring much memory burden because of the fine-grained modeling.

*3.5.2 Time Complexity.* We extract the visual features of clips in the data preparation step, which will not increase time complexity in the process of model training. Because our model is made up of graph convolutional layers based on the user-clip interactions, the time complexity mainly derives from the embedding propagation on dual-side user-clip relation graphs. For example, in the positive relation graph, first we conduct an embedding propagation step to maintain user embeddings , and the time cost is $O(\|\tilde{\mathbf{R}}^p\|_0 d + Md^2)$, where $\|\tilde{\mathbf{R}}^p\|_0$ is number of nonzeros in the positive interaction matrix $\tilde{\mathbf{R}}^p$, $M$ is number of users and $d$ is the dimension of embeddings. Next a two-hop embedding propagation is conducted to capture the higher-order information transmission between clips, the time cost of which is $O(\|\tilde{\mathbf{R}}^p\|_0 d + Cd^2)$. Therefore, the time cost on positive-side graph convolution is $O(2\|\tilde{\mathbf{R}}^p\|_0 d + Md^2 + Cd^2)$. Similarly, the time cost on negative-side graph convolution is $O(2\|\tilde{\mathbf{R}}^n\|_0 d + Md^2 + Cd^2)$. The total time cost is $O(2\|\tilde{\mathbf{R}}^p\|_0 d + 2\|\tilde{\mathbf{R}}^n\|_0 d + 2Md^2 + 2Cd^2)$. It can be seen that we decrease the time complexity compared with traditional GCN propagation by distinguishing the embedding propagation according to node types and decreasing the size of adjacency matrix from $(M+C)\times(M+C)$ to $M\times C$. In conclusion, our model will not take more time than existing GCN models.

# 4 EXPERIMENT

In this section, we carefully collect two real-world datasets and conduct experiments to answer the following research questions.

- **RQ1:** How does the proposed method perform compared with state-of-the-art models for micro-video recommendation?
- **RQ2:** How do the fine-grained modeling of video features and user preferences influence our model's effectiveness?
- **RQ3:** How do the hyper-parameters settings affect the final performance of our model?
- **RQ4:** Can the proposed method capture the evolution of fine-grained user interest during the video playing process effectively?

## 4.1 Experimental Settings

*4.1.1 Dataset.* We evaluate the recommendation performance on two large-scale micro-video datasets both from one of the largest micro-video platforms. Specifically, two datasets are collected from different mobile Apps, covering different users and videos. [3] The datasets include both the users' fine-grained feedback and the original video data, which has not been covered by any public data. Table 1 summarizes the basic statistics of the two datasets.

[3]The data collection strictly follows both user privacy protection and commercial regulations, confirmed by both involved users and the company.

**Table 2: Statistics of our collected datasets.**

| Dataset | #Users | #Videos | #Interactions |
|---|---|---|---|
| Micro-video-A | 12,739 | 58,291 | 342,694 |
| Micro-video-B | 22,049 | 61,903 | 2,111,566 |

- **Micro-video-A.** This dataset is collected from one specific version of the platform's mobile App. Each interaction record includes user-id, video-id, the user's playing time, duration time of the video, interaction timestamp and multi-level behaviors including like, follow, and forward. The playing time and duration time are used to calculate the skip time of the user. Besides, we have collected all the micro-video files that appeared in the interaction record to extract visual features.
- **Micro-video-B.** This dataset is collected from another mobile App of the same platform, which includes more interaction records than Micro-video-A dataset. The data fields are similar to Micro-video-A, but it has **different users and videos** on a **different App** and thus can be regarded as a totally-different dataset.

*4.1.2 Evaluation Metrics.* To evaluate the performance of each model, we use four widely adopted metrics including AUC, Logloss, Recall and NDCG, which are defined as follows:

- **AUC** indicates the probability that the positive sample's score is higher than the negative counterparts, reflecting the quality of the model's discriminating ability.
- **Logloss** measures the gap between the prediction probability score and the ground-truth, which incarnates the accuracy on an absolute level.
- **Recall@K** measures the ratio of test items that have been successfully recommended in the top-K ranking list. We set K as 3 and 5, which are both widely-used settings in existing works [1, 9].
- **NDCG@K** assigns higher scores to hits at higher positions in the top-K ranking list, which emphasizes that the positive samples should be ranked as higher as possible.

*4.1.3 Baselines.* To demonstrate the effectiveness of our FRAME model, we make comparisons with competitive methods suitable for the micro-video recommendation task. The baselines are classified into two categories: state-of-the-art micro-video recommendation models and feature-based recommendation models, which are also known as CTR prediction models, that can well leverage complex features. To make a fair comparison for feature-based methods, we feed the same video visual feature as our model, which makes the models work on a fine-grained level as well.

**Micro-video recommender Models:**

- **ALPINE** [27]: This method aims to model user's dynamic and diverse interests, which utilizes a temporal graph-guided LSTM network connecting the videos with similar visual features. What's more, this method learns the representation of users by considering multi-level user interests.
- **MTIN** [23]: This method concentrates on addressing the multi-scale time effect on user interests and constructs user interest groups based on user interaction sequences to model diverse user interests. It achieves state-of-the-art performance in micro-video recommendation.

**Table 3: Overall performance comparison (all models have the same input for fair comparison; values are the average of three running instances with different random seeds; best baseline is marked with underline).**

| Method Type | Method | Micro-video-A | | | | | | Micro-video-B | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AUC | Logloss | Recall@3 | NDCG@3 | Recall@5 | NDCG@5 | AUC | Logloss | Recall@3 | NDCG@3 | Recall@5 | NDCG@5 |
| Feature-based recommender model | DeepFM | 0.5834 | 0.8675 | 0.0563 | 0.1975 | 0.0832 | 0.2348 | 0.6566 | 0.8319 | 0.0785 | 0.2498 | 0.1064 | 0.3172 |
| | NFM | 0.6049 | 0.7835 | 0.0682 | 0.2374 | 0.0886 | 0.2591 | 0.6673 | 0.8125 | 0.0876 | 0.2631 | 0.1296 | 0.3319 |
| | AutoInt | 0.6279 | 0.7931 | 0.0694 | 0.2244 | 0.0907 | 0.2652 | 0.6865 | 0.7847 | 0.0842 | 0.2653 | 0.1287 | 0.3574 |
| | DIFM | 0.6338 | 0.7531 | 0.0749 | 0.2448 | 0.1104 | 0.3086 | 0.6913 | 0.7957 | 0.0861 | 0.2710 | 0.1256 | 0.3724 |
| | AFN | 0.6376 | 0.7438 | 0.0762 | 0.2683 | 0.1085 | 0.3106 | 0.6808 | 0.8045 | 0.0903 | 0.2794 | 0.1376 | 0.3830 |
| Micro-video recommender model | ALPINE | 0.6218 | 0.7692 | 0.0701 | 0.2324 | 0.0925 | 0.2886 | 0.6995 | 0.7834 | 0.0922 | 0.2748 | 0.1320 | 0.3783 |
| | MTIN | 0.6427 | 0.7228 | 0.0836 | 0.2715 | 0.1125 | 0.3327 | 0.7489 | 0.7647 | 0.1036 | 0.3081 | 0.1417 | 0.3938 |
| Our model | **FRAME** | **0.7039** | **0.6732** | **0.1083** | **0.3219** | **0.1378** | **0.3722** | **0.7870** | **0.7619** | **0.1149** | **0.3296** | **0.1796** | **0.4358** |

**Feature-based Recommender Models:** We compared our method with five competitive feature-based recommender models shown as follows, to make a fair comparison for the methods, we feed the same video visual feature as our model, which makes these models work on a fine-grained level as well.

- **DeepFM** [17]: This method combines LR and FM to model the second-order feature interaction.
- **AutoInt** [35]: This method introduces self-attention networks to model high-order feature interactions.
- **NFM** [19]: This method uses FM and neural network to model second and higher-order feature interactions.
- **DIFM** [30]: This method combines the bit-wise and vector-wise feature representation to learn more flexible representations for given features.
- **AFN** [7]: This method takes logarithmic transformation layers to learn adaptive-order feature interactions.

**Discussion of baseline selection.** We make careful consideration of the baselines for comparison. As for feature-based methods, we use the stable and effective ones; for micro-video recommender models, we choose the state-of-the-art methods with attention to user interest modeling. It is worth mentioning that some other recommendation models [26, 39, 42] using micro-video datasets are not appropriate for comparison. For example, some works like [26] focus on the cross-domain recommendation which is different from our task definition. Some other methods aim to explore the modality-specific recommendation [39, 42], which share rare similarities with us on the utilization of video content.

*4.1.4 Hyper-parameter Settings.* Our model is implemented in PyTorch. We use Adam [24] for optimization with the initial learning rate as 0.001. The batch size is set as 1024 and the embedding size is 128 for all models, following existing works [23]. Xavier initialization [16] is used to initialize the parameters. We use the extracted visual video features as the input of all compared methods. We carefully tune the hyper-parameter setting for all baselines following the original papers' settings or suggestions. In our model, the number of clips for each video $N_c$ is carefully searched in [1, 4, 8], based on the limit of our computation resources. The ratio of the positive and negative result weight $\alpha_p : \alpha_n$ is carefully searched in [1:0.1, 1:0.2, 1: 0.3, 1:0.4, 1:0.5]. The ratio of the point-wise loss and pair-wise loss $\alpha : \beta$ is searched in [0.2:1, 0.5:1, 1:1, 1:0.2, 1:0.5]. The L2 normalization coefficient $\lambda$ is searched in [1e-6, 1e-5, 1e-4].

## 4.2 Overall Performance (RQ1)

From the results in Table 3, we have the following observations:

- **Our proposed method consistently achieves the best performance compared with baselines.** We can observe that our model FRAME consistently outperforms all baselines in all metrics. Specifically, on Micro-video-A our model improves AUC by 9.52% compared with the best baseline MTIN, average 26.02% for Recall, 15.22% for NDCG and improves Logloss by 6.86%. As for Micro-video-B, our model improves AUC by 5.09% compared with the best baseline MTIN, average 18.82% for Recall, 8.59% for NDCG and decreases Logloss by 0.37%. The improvement is more obvious on Micro-video-A in which users are not so active. In this situation fine-grained user modeling can aggregate more information than previous methods.
- **Fine-grained user interest modeling indeed improves the model performance.** ALPINE and MTIN are two effective baselines in micro-video recommendation, while their designs still stay on the video-level. In comparison our model utilizes a fine-grained user-clip interaction manner to learn users' fine-grained positive and negative interests. On the one hand, it weakens the worse performance caused by the insufficiency of historical interactions. On the other hand, it considers the variety of user interests in one video, enhancing the representation power of user embedding. Intuitively, the comparison shows that our fine-grained user interest modeling achieves better performance.
- **Fine-grained video features support to improve the performance.** In the comparison experiments, we perform the fine-grained clipping on the videos for the feature-based methods as well. The result shows that some methods outperform the micro-video recommender models, indicating that the obtained fine-grained video features benefit the performance. For example, AFN gets better performance on all metrics than ALPINE on two datasets. While in FRAME, we have more ingenious designs which guide to learn better features for users and clips from the fine-grained user-video interactions. This further improves the performance of our model compared with feature-based methods.

## 4.3 Ablation Study (RQ2)

*4.3.1 Effectiveness of fine-grained clip dividing on videos.* In our model design, we divide each video into $N_c$ clips and extract the visual feature of each clip, which reflects the fine-grained modeling of both video features and user preferences. We compare the performance of our model without dividing the video ($N_c = 1$) and dividing the video into 4 and 8 clips, as shown in Table 4.

**Table 4: Ablation study of the fine-grained clip dividing.**

| Dataset | Model | AUC | Logloss | Recall@3 | NDCG@3 |
|---|---|---|---|---|---|
| Micro-video-A | w/o clip dividing ($N_c = 1$) | 0.6329 | 0.7658 | 0.0736 | 0.2715 |
| | w/ clip dividng ($N_c = 4$) | 0.6967 | 0.6903 | 0.1058 | 0.3145 |
| | w/ clip dividng ($N_c = 8$) | **0.7039** | **0.6732** | **0.1083** | **0.3219** |
| Micro-video-B | w/o clip dividing ($N_c = 1$) | 0.7318 | 0.8054 | 0.0910 | 0.2684 |
| | w/ clip dividng ($N_c = 4$) | 0.7739 | 0.7715 | 0.1127 | 0.3173 |
| | w/ clip dividng ($N_c = 8$) | **0.7870** | **0.7619** | **0.1149** | **0.3296** |

**Table 5: Ablation study of the treatment on visual features.**

| Dataset | Model | AUC | Logloss | Recall@3 | NDCG@3 |
|---|---|---|---|---|---|
| Micro-video-A | w/o visual features | 0.6318 | 0.7435 | 0.0677 | 0.2648 |
| | w/ original visual features | 0.6628 | 0.7268 | 0.0946 | 0.2895 |
| | w/ enhanced visual features | **0.7039** | **0.6732** | **0.1083** | **0.3219** |
| Micro-video-B | w/o visual features | 0.7035 | 0.8255 | 0.0898 | 0.2739 |
| | w/ original visual features | 0.7642 | 0.7936 | 0.1026 | 0.3077 |
| | w/ enhanced visual features | **0.7870** | **0.7619** | **0.1149** | **0.3296** |

**Table 6: Ablation study about introducing negative clips into the model.**

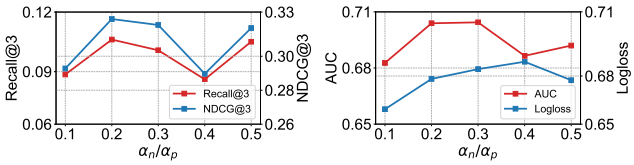
| Dataset | Model type | AUC | Logloss | Recall@3 | NDCG@3 |
|---|---|---|---|---|---|
| Micro-video-A | full model | **0.7039** | **0.6732** | **0.1083** | **0.3219** |
| | w/o negative clips | 0.6019 | 0.7613 | 0.0628 | 0.2069 |
| Micro-video-B | full model | **0.7870** | **0.7619** | **0.1149** | **0.3296** |
| | w/o negative clips | 0.6724 | 0.8429 | 0.0812 | 0.2696 |

It is shown that the more clips divided for the video, the better performance achieved by the model. The gap between the model without clip dividing and with 4 clips for each video is especially obvious. On average we get 7.9% improvement on AUC, 7.1% on Logloss, 33.8% on Recall@3 and 17.0% on NDCG@3 respectively by using clips rather than the full video on the two datasets, which demonstrates the significance of using more clips in our model. While the difference between $N_c = 4$ and $N_c = 8$ shrinks. Since setting $N_c > 8$ will cause larger computation cost while a tiny improvement, we do not choose larger values here.

*4.3.2 Effectiveness of the treatment on visual features.* We first conduct visual feature extraction in our method and then introduce an enhanced layer to encourage the clip embedding to express more information. Here we explore the effect of different treatments on visual clip features including without visual features, with original extracted visual features and with enhanced visual features. The performance comparison is shown in Table 5.

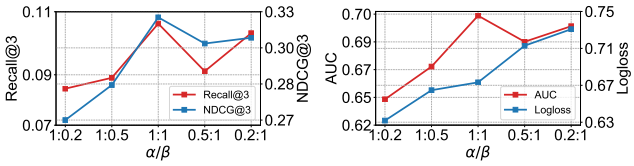
It can be seen that the model without using the visual feature learning from user-clip interaction can't catch up with the model with extracted visual features. To be specific, we get improvement on the two datasets by 6.8% on AUC, 3.1% on Logloss, 26.9% on Recall@3 and 10.8% on NDCG@3. In addition, the enhanced clip embedding can further improve performance, which indicates the necessity of the feature transformation operation.

*4.3.3 Effectiveness of introducing negative clips into the user interest modeling.* One of the key designs in fine-grained user interest modeling is introducing negative clips to learn user interest representation. Here we compare the performance of the full model and




(a) Recall and NDCG (b) AUC and Logloss

**Figure 4: Study of negative/positive ratio in prediction on Micro-video-A dataset.**




(a) Recall and NDCG (b) AUC and Logloss

**Figure 5: Study of the ratio of point-wise loss and pair-wise loss on Micro-video-A dataset.**

the version without using negative clips, which means removing the message-passing through negative relation graph and no use of the pair-wise loss term. The result is shown in Table 6, from which it can be found that introducing negative clips can bring significant improvements (37.04% in average). Therefore, it's of great significance to consider the negative feedback in the micro-video recommendation, which serves as an effective signal complementing with the positive feedback.

## 4.4 Hyper-parameter Study (RQ3)

As our model takes the combination form of dual-side sets to give prediction and the hybrid loss function with weight, in this section we investigate the influence of settings on these two parts to the performance of our model on Micro-video-A as presented in Figure 4-5.

First we explore the impact of the ratio between weights of positive-side and negative-side sets for the clip-level prediction. We evaluate the ratio of the negative and positive prediction weight $\alpha_n : \alpha_p$ in [0.1, 0.2, 0.3, 0.4, 0.5]. In Micro-video-A, we can observe that the best performance on AUC, Recall and NDCG appears when the ratio is 0.2. The best Logloss is achieved when the ratio is 0.1. We choose 0.2 as the final hyper-parameter setting on this dataset.

Next we conduct experiments on how the ratio of two loss functions' weight $\alpha : \beta$ influences our model's effectiveness. We compare the performance when this ratio is traversed in [0.2:1, 0.5:1, 1:1, 1:0.5, 1:0.2]. Intuitively from the result we can find that as the weight of the point-wise loss decreases, the Logloss rises. The other three metrics get the best results at the same time when the ratio reaches 1:1 on Micro-video-A, where the model maintains the best performance. Therefore, we use 1:1 as the final setting of the dataset.

## 4.5 Case Study (RQ4)

Here we conduct experiments to verify the ability to capture fine-grained user interest during the playing process of videos. A visualization example is shown in Figure 6. We take one interaction

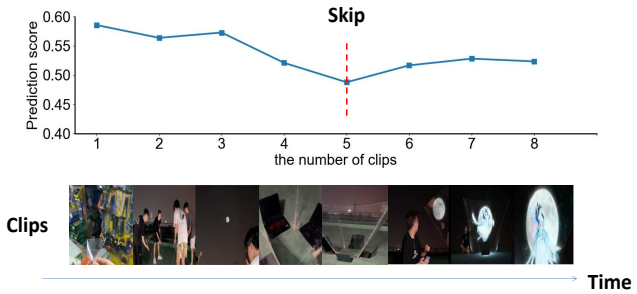



**Figure 6: A visualization of capturing fine-grained evolution of user interest between clips in one video.**

record from the test set of Micro-video-B, in which videos are divided into 8 clips. The figure above displays the clip-level prediction score given by our model. It can be seen that the user preference varies between clips in the same video, which gets the lowest score at the 5-th clip. Meanwhile we calculate the ratio the user watched of the video, that is 0.584. We can easily find that the skipping moment falls in the 5-th clip exactly. The figure below shows the content of different clips of this video, which actually differs from each other. A reasonable explanation is that users keep watching until the content makes them intolerable or bored. Intuitively it verifies the ability to capture fine-grained user interest via learning the continuous user feedback in this scenario, which meets the expectation in our initial motivation.

# 5 RELATED WORK

## 5.1 Micro-video Recommendation

Generally speaking, the existing methods of micro-video recommendation can be classified into three categories: collaborative filtering methods [19, 22, 36, 37], content-based methods [5, 8, 10, 38, 40] and hybrid ones [3, 45]. Collaborative filtering methods models user interests assuming that users with similar interests tend to make similar decisions. Wang et al. [36] described user-item interactions in a graph network and adopted sequential recommendation paths to model the user-item correlations. While there are disadvantages for CF-based methods due to the cold-start problem and data sparsity, leading to the performance limitation of these methods [12, 29, 36]. In order to address these problems, content-based methods are developed, which measure the similarity between the target video and the videos in user interactions. These approaches regard user interest modeling as a common issue when handling user-video interactions. For instance, Chen et al. [4] made use of category-level and item-level attention mechanisms to model user's diverse interests. As for hybrid methods, they tend to integrate two types of methods mentioned above into one framework.

In spite of the remarkable advancement of the performance, almost all existing methods learn user interest taking the entire video as a unit, which is not capable of capturing fine-grained user interest towards different parts of the video. Different from them, we first conduct fine-grained clip segmentation on the videos and then exploit the user-clip interactions via a brand new manner to capture the fine-grained user interest.

## 5.2 Feature-based Recommendation

Click-through rate (CTR) prediction, which goal is to predict the probability of a user clicking on an ad or an item, is of great significance to many online applications such as online advertising and recommender systems [6, 11, 13, 15, 21]. Machine learning has been widely used in click-through rate prediction, which is usually formulated as a supervised learning task with user profiles and item attributes as input features. An earlier influential CTR prediction scheme is Factorization Machines (FM) [33] which combines SVM with factorization methods to model feature interactions. While it can only capture the low-order interaction but fail to model the high-order feature interaction. As the growing and widely application of deep learning, many methods based on deep neural networks [2, 6, 7, 17, 19, 32, 35, 43, 44] are proposed to model high-order interaction. For example, AutoInt [35] uses self-attention networks to learn high-order feature interactions, in which different orders of feature combinations of input features can be considered.

Our model aims to predict the probability a user finishes watching a given video, which is essentially finding representations of input features and modeling the interactions between features. Therefore methods for CTR prediction can also be adopted here.

# 6 CONCLUSIONS AND FUTURE WORK

In this work, we approach the problem of micro-video recommendation from a brand new perspective, learning from fine-grained user feedback, which is critical but cannot be supported by public datasets. In the real-world scenario, the user feedback in micro-video recommendation is presented in a continuous form, which reveals fine-grained user preferences. While existing methods still model user interest at a coarse-grained level which is limited in learning fine-grained user preference. To address these problems, we first carefully collect two real-world large-scale datasets. Then we propose a method named FRAME, which first transforms the pre-trained visual features, then employs graph convolutional layers to learn from complex user-clip interactions from both positive and negative side, and finally supervises the training process by a hybrid loss function. Experiments show the significant performance improvement of FRAME and verify the effectiveness of fine-grained user modeling.

In the future, we plan to conduct an online A/B test to evaluate the proposed method. We also plan to conduct experiments on more powerful computation clusters with finer-grained video processing by increasing the granularity of clips. Besides, we will encode the temporal relations between clips to improve the work further.

# ACKNOWLEDGMENTS

This work is supported in part by The National Key Research and Development Program of China (No. 2022ZD0117900), the National Natural Science Foundation of China (No. 62272262, No. 61972223, No. U1936217, No. U20B2060, No. U20B2062) and the Fellowship of China Postdoctoral Science Foundation (No. 2021TQ0027, No. 2022M710006).

## REFERENCES


[1] Ye Bi, Liqiang Song, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. DCDIR: a deep cross-domain recommendation system for cold start users in insurance domain. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1661–1664.

[2] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. 2022. CAN: Feature Co-Action Network for Click-Through Rate Prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 57–65.

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.

[4] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou, Zhiwei Xiong, and Yan Li. 2018. Temporal hierarchical attention at category-and item-level for micro-video click-through prediction. In *MM*. 1146–1153.

[5] Xusong Chen, Rui Zhao, Shengjie Ma, Dong Liu, and Zheng-Jun Zha. 2018. Content-based video relevance prediction with second-order relevance and attention modeling. In *Proceedings of the 26th ACM international conference on Multimedia*. 2018–2022.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[7] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.

[8] 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.

[9] Paolo Cremonesi, Primo Modica, Roberto Pagano, Emanuele Rabosio, and Letizia Tanca. 2015. Personalized and context-aware TV program recommendations based on implicit feedback. In *International Conference on Electronic Commerce and Web Technologies*. Springer, 57–68.

[10] Peng Cui, Zhiyu Wang, and Zhou Su. 2014. What videos are similar with you? learning a common attributed representation for video recommendation. In *Proceedings of the 22nd ACM international conference on Multimedia*. 597–606.

[11] Wei Deng, Junwei Pan, Tian Zhou, Deguang Kong, Aaron Flores, and Guang Lin. 2021. DeepLight: Deep lightweight feature interactions for accelerating CTR predictions in ad serving. In *Proceedings of the 14th ACM international conference on Web search and data mining*. 922–930.

[12] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 515–524.

[13] Zhifang Fan, Dan Ou, Yulong Gu, Bairan Fu, Xiang Li, Wentian Bao, Xin-Yu Dai, Xiaoyi Zeng, Tao Zhuang, and Qingwen Liu. 2022. Modeling Users' Contextualized Page-wise Feedback for Click-Through Rate Prediction in E-commerce Search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 262–270.

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

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

[16] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[17] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

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

[19] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[20] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[21] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–9.

[22] Yanxiang Huang, Bin Cui, Jie Jiang, Kunqian Hong, Wenyu Zhang, and Yiran Xie. 2016. Real-time video recommendation exploration. In *Proceedings of the 2016 International Conference on Management of Data*. 35–46.

[23] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-scale Time-aware User Interest Modeling for Micro-video Recommendation. In *MM*. 3487–3495.

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

[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[26] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3161–3171.

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

[28] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-Aware Message-Passing GCN for Recommendation. In *Proceedings of the Web Conference 2021*. Association for Computing Machinery, 1296–1305.

[29] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-video co-attention network for personalized micro-video recommendation. In *The World Wide Web Conference*. 3020–3026.

[30] Wantong Lu, Yantao Yu, Yongzhe Chang, Zhen Wang, Chenhui Li, and Bo Yuan. 2021. A dual input-aware factorization machine for CTR prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 3139–3145.

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

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

[33] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[35] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[36] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.

[37] Pengfei Wang, Hanxiong Chen, Yadong Zhu, Huawei Shen, and Yongfeng Zhang. 2019. Unified collaborative filtering over graph embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 155–164.

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

[39] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, Liqiang Nie, and Min Zhang. 2021. DualGNN: Dual Graph Neural Network for Multimedia Recommendation. *IEEE Transactions on Multimedia* (2021).

[40] Yinwei Wei, Xiang Wang, Weili Guan, Liqiang Nie, Zhouchen Lin, and Baoquan Chen. 2019. Neural multimodal cooperative learning toward micro-video understanding. *IEEE Transactions on Image Processing* 29 (2019), 1–14.

[41] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM international conference on multimedia*. 3541–3549.

[42] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.

[43] Kai Zhang, Hao Qian, Qing Cui, Qi Liu, Longfei Li, Jun Zhou, Jianhui Ma, and Enhong Chen. 2021. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 984–992.

[44] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*. Springer, 45–57.

[45] Xiaojian Zhao, Guangda Li, Meng Wang, Jin Yuan, Zheng-Jun Zha, Zhoujun Li, and Tat-Seng Chua. 2011. Integrating rich information for video recommendation with multi-task rank aggregation. In *Proceedings of the 19th ACM international conference on Multimedia*. 1521–1524.