

# Click-Through Rate Prediction with Multi-Modal Hypergraphs

## ABSTRACT

Advertising is critical to many online e-commerce platforms such as e-Bay and Amazon. One of the important signals that these platforms rely upon is the click-through rate (CTR) prediction. The recent popularity of multi-modal sharing platforms such as TikTok has led to an increased interest in online micro-videos. It is, therefore, useful to consider micro-videos to help a merchant target micro-video advertising better and find users' favourites to enhance user experience. Existing works on CTR prediction largely exploit unimodal content to learn item representations. A relatively minimal effort has been made to leverage multi-modal information exchange among users and items. We propose a model to exploit the temporal user-item interactions to guide the representation learning with multi-modal features, and further predict the user click rate of the micro-video item. We design a Hypergraph Click-Through Rate prediction framework (HyperCTR) built upon the hyperedge notion of hypergraph neural networks, which can yield modal-specific representations of users and micro-videos to better capture user preferences. We construct a time-aware user-item bipartite network with multi-modal information and enrich the representation of each user and item with the generated interests-based user hypergraph and item hypergraph. Through extensive experiments on three public datasets, we demonstrate that our proposed model significantly outperforms various state-of-the-art methods.

## 1 INTRODUCTION

Click-Through Rate (CTR) prediction has become one of the core components of modern advertising on many e-commerce platforms. The goal is to predict customers' click probability on wide range of items. Existing works on CTR prediction only focus on modeling pairwise interactions from uni-modal features which might not lead to satisfactory results. This existing gap leads to new opportunities where we can exploit the widely available multi-modal features which is largely unexplored. Besides, they can given complementary information to the model which alone cannot be obtained via uni-modal modeling. AutoFIS [19] and UBR4CTR [26] are recent Factorization Machine (FM) [31] based models with multi-layer perceptron (MLP) which mainly utilize user-item interactions features. To supplement the lack of additional information, deep neural networks (DNNs) are also explored with automated feature engineering. For example, DSTN [24] leverages DNNs-based method to fuse additional auxiliary data and item information to further uncover hidden information. Although these representative works have achieved good performance, there are still limited exploration on modeling multi-modal features and how they could contribute towards the model performance.

Recently, the wide-spreading influence of micro-video sharing platforms, e.g., Tiktok<sup>1</sup> and Kuaishou<sup>2</sup> make them a popular platform for socialising, sharing and advertising as micro-videos. These videos are compact and come with rich multimedia content from

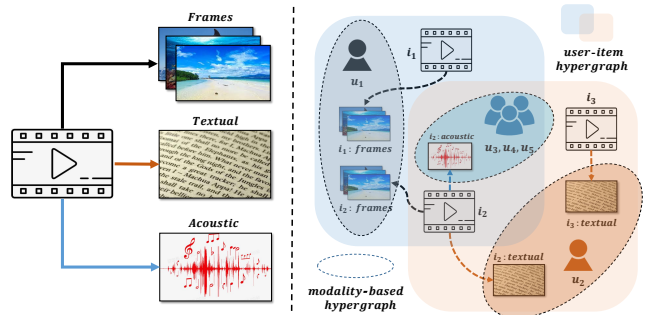


Figure 1: An illustration of multi-modal user preferences.

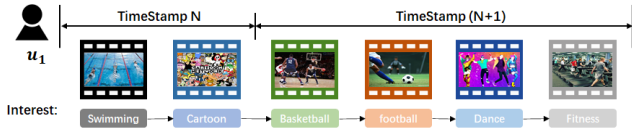
multiple modalities, i.e., textual, visual, as well as acoustic information. Motivated by this, we propose a novel method which addresses the limitations in current methods and improve CTR prediction performance through micro-videos. However, modeling multi-modal features extracted from micro-videos for CTR prediction in a holistic way is not straightforward. First, in a typical setting of CTR prediction, the interactions between users and items are normally sparse, and the sparsity issue becomes even more severe (in magnitude of number of modalities) when taking into account multi-modal features. For example, compared to uni-modal feature space, the sparsity of a dataset is tripled when considering visual, acoustic and text features of a target item. Therefore, effectively mitigating the sparsity issues introduced by multi-modal features without compromising upon the performance of the model is the key to this problem.

We rely on hypergraphs to address some of the challenges. Hypergraph [3] extends the concept of an edge in a graph and can connect more than two nodes. Inspired by the flexibility and expressiveness of hypergraphs, we use the concept to multi-modal feature modeling, and propose a new model based on modality-originated hypergraphs by which the sparsity issues between users and items under each modality can be alleviated. Figure 1 is an example of the proposed modality-originated hypergraphs, where user  $u_1$  and user  $u_2$  both have interactions with multiple micro-videos, e.g.,  $i_1$  and  $i_2$ , in which each hyperedge can connect multiple item nodes on a single edge. Compared with a simple graph on which the degree of all edges is set to be 2, a hypergraph can encode high-order data correlation (beyond pairwise connections) using its degree-free hyperedges. Different from various modalities, we incorporate different multi-modal information, e.g., frames, acoustic, textual into user-item hypergraphs to help establish an in-depth understanding of user preferences. The reason for considering using hypergraphs in our work is due to the purpose of building modality-originated hypergraphs which can be treated as data argumentation technique.

We also construct hypergraphs considering both user and item. In Figure 1, user  $u_1$  cares more about frames of micro-video  $i_2$ , whereas user  $u_2$  might be fond of the text content. Hence, different users might have different tastes on modalities of a micro-video. A

<sup>1</sup><https://www.tiktok.com/>

<sup>2</sup><https://www.kuaishou.com/>



**Figure 2: Illustration of user  $u_1$ 's historical view records with micro-videos, which reflects the user's global view interests.**

group of users  $u_3$ ,  $u_4$  and  $u_5$  click micro-video  $i_2$  due to the intriguing sound tracks. Such signals can be utilized to construct a group-aware hypergraph which is comprised of multiple users who share the same interest for the item. Inspired by the recent success of self-supervised learning (SSL) [20], we utilize the mutual information maximization principle to learn the intrinsic data correlation [40] to help construct the interests-based hypergraph where we represent a group of users with common preference on modal-specific content. Hence, in each modality (e.g., visual), we aggregate information from the group-aware hypergraph and incorporate them into user representations. According to group-aware hypergraph, each user has interactions with one of the item's modalities, while different items can be interacted with the same user. For example, user  $u_1$  likes  $i_1$ 's frames, and  $u_1$  will pay more attention to the visual-aspect of other items. Under such circumstances, we can also construct a homogeneous item-level hypergraph comprising of multiple items who have certain potential modality that appeal to the same user.

Generally, user preference evolves over time, and it is hence a sequential phenomenon. As shown in Figure 2, user  $u_1$  has watched swimming and cartoon videos at timestamp  $N$ , indicating that the user has two very different interests and we cannot capture the user's interests at the single time point. If at a new timestamp  $N + 1$ , basketball, football, dance and fitness videos have selected by the same user. Then, we can infer that this user has more interests in sports than comedy. Under such circumstances, some researches consider users' interest as dynamic when designing CTR systems and have better model users' interest such as THACIL [4]. Therefore, more user-behavior modeling methods are proposed for tackling this problem. There are RNN-based models [13, 17], CNN-based models [33], transformer-based models [25] and memory network-based models [7].

To tackle the aforementioned problem, we propose HyperCTR, a novel temporal framework with user and item level hypergraphs to enhance CTR prediction. To explore the sequential correlations at different time slots, HyperCTR truncates the user interactions based on the timestamp to construct a series of hypergraphs. With a hypergraph convolutional network (HGCN), HyperCTR can aggregate the correlated users and items with direct or high-order connections to generate the dynamic embedding at each time slot. With change happening both over time and across users, the temporal and group-aware user embeddings is fed into a fusion layer to generate the final user representation. The prediction of an unseen interaction can be calculated as probability between the user and micro-video representations after MLP. We show the effectiveness of our framework on three publicly available datasets – Kuaishou, Micro-Video 1.7M (MV1.7M) and MovieLens. Our **key contributions** are: 1) We study the dynamics of user preference from two perspectives - time-aware

and group-aware - and uncover the importance in exploiting the information interchange on various modalities to reflect user interests and affect CTR performance. 2) We propose a novel method HyperCTR framework with two types of modality-originated hypergraphs to generate users and items embeddings. Three of the unique aspects of the framework are a self-attention layer to capture the dynamic pattern in user-item bipartite interaction networks, a fusion layer to encode each interaction with both the temporal individual embeddings and group-level embeddings for final user pattern modeling and the CTR probability will be calculated by a MLP layer with the input of user- and item-level embeddings. 3) Extensive experiments on three public datasets demonstrate that our proposed model outperforms several state-of-the-art models. Due to anonymous requirements, the code link is invisible until paper acceptance.

## 2 OUR NOVEL HYPERCTR MODEL

In this section, we present our novel click-through rate prediction framework with multi-modal hypergraphs.

### 2.1 Preliminaries

Our goal is learning user preferences from the hypergraph structure and predicting the probability that a user clicks the recommended entities. We denote  $U$  to represent the set of users and  $I$  represents the set of  $P$  items in an online platform. The item is characterised by various modalities, which are visual, acoustic, and textual. We also have historical interactions, such as “view”, “like”, “follow” and “click” between users and items. We represent this interaction as a hypergraph  $\mathcal{G}(u, i)$ , where  $u \in U$  and  $i \in I$  separately denote the user and item sets. A hyperedge,  $\mathcal{E}(u, i_1, i_2, i_3, \dots, i_n)$  indicates an observed interaction between user  $u$  and multiple items  $(i_1, i_2, i_3, \dots, i_n)$  where hyperedge is assigned with a weight by  $\mathbf{W}$ , a diagonal matrix of edge weights. We also have multi-modal information associated with each item, such as visual, acoustic and textual features. For instance, we denote  $M = \{v, a, x\}$  as the multi-modal tuple, where  $v$ ,  $a$ , and  $x$  represent the visual, acoustic, and textual modalities, respectively.

Our hypothesis is that user preference also plays an important role. A user group  $y$  is associated with a user set  $C_y \in U$  which can be used to represent a  $N$ -dimensional group-aware embedding. The member of groups might change over time. For each user  $u$ , we denote the user's temporal behavior as  $B_u^c$  responding to the current time, and sequential view user behavior as  $B_u^s$  according to a time slot. We further utilize  $\mathcal{K}(B_u^c)$  and  $\mathcal{K}(B_u^s)$  to represent the set of items in the sequential behavior, respectively.

We explain some important terminologies below which will help build a foundation to understand the technical merits of our work which includes temporal user-item interaction representation, group-aware hypergraph and item hypergraphs.

- *Definition 1 (Temporal User-item Interaction Representation)*  
Let a sequence  $\mathcal{S}(u, i_1, i_2, i_3, \dots)$  indicate an observed interaction between user  $u$  and multiple items  $(i_1, i_2, i_3, \dots)$  occurring during a time slot  $t_n$ . We denote  $\mathbf{E}_I = [\mathbf{e}_1, \mathbf{e}_2, \dots]$  as the set of items' static latent embeddings, which represent the set of items a user interacts with during this time slot. Each item in current sequence is associated with multi-modal features,

which utilize  $M_{i_n}$  and it contains three-fold information about visual, acoustic and textual, denoted as  $v_{i_n}$ ,  $a_{i_n}$  and  $x_{i_n}$ , respectively.

- **Definition 2 (Group-aware Multi-Modal Hypergraph)**  
Let  $\mathcal{G}_g^{t_n}$  represent a hypergraph associated with  $i$ -th item at time slot  $t_n$ .  $\mathcal{G}_g^{t_n} = \{V_g^{t_n}, \mathcal{E}_g^{t_n}, \mathbf{W}_g^{t_n}, \mathbf{H}_g^{t_n}\}$  is constructed based on the whole user-item interactions with multi-modal information.  $V_g^{t_n}$  represents the nodes of individual and the correlated item in  $\mathcal{G}_g^{t_n}$ ,  $\mathcal{E}_g^{t_n}$  denoted as a set of hyperedges. We are thus creating a link to users who have interactions with multiple modal list of items. Each  $\mathcal{G}_g^{t_n}$  is associated with an incidence matrix  $\mathbf{H}_g^{t_n}$  and it is also associated with a matrix  $\mathbf{W}_g^{t_n}$ , which is a diagonal matrix representing the weight of the hyperedge  $\mathcal{E}_g^{t_n}$ .
- **Definition 3 (Item Homogeneous Hypergraph)**  
There are three hyperedges in each  $\mathcal{G}_i^{t_n}$ , which was defined in Definition 2. Let  $\mathcal{G}_i^{t_n} (\mathcal{G}_i^{t_n} \supseteq \{\mathbf{g}_v^{t_n}, \mathbf{g}_a^{t_n}, \mathbf{g}_x^{t_n}\})$  represent a series of item homogeneous hypergraphs for each user group member.  $\mathcal{G}_i^{t_n} = \{V_i^{t_n}, \mathcal{E}_i^{t_n}, \mathbf{W}_i^{t_n}, \mathbf{H}_i^{t_n}\}$  is constructed based on each  $\mathcal{G}_i^{t_n}$  and describes a set of items that a user interacts with generated in the time slot  $t_n$ .  $V_i^{t_n}$  represents the nodes of items and  $\mathcal{E}_i^{t_n}$  denotes a set of hyperedges, which is creating the link to items which have interactions with a user.

The group-aware hypergraph capture group member’s preference, while item hypergraphs pay more attention to item-level high-order representation. Two types of hypergraphs are the fundamental for our temporal user-item interaction representation. We define our multi-modal hypergraph CTR problem as follow:

- **Problem 1 Click-Through Rate Prediction** Given a target user intent sequence  $\mathcal{S}$ , and its group-aware hypergraph  $\mathcal{G}_g^{t_n}$  and item hypergraph  $\mathcal{G}_i^{t_n}$ , both of them depending on the time sequence  $T$ , this problem can be reformulated as a function  $f(u, \mathcal{G}_g^{t_n}, \mathcal{G}_i^{t_n}, i) \rightarrow y$  for a recommended item  $i$ , where denotes  $y$  the probability that user clicks or not.

## 2.2 HYPERCTR Framework

HyperCTR framework is illustrated in Figure 3. The framework can be divided into four components, which are, temporal user behavior attention module, interests-based user hyperedge generation module, item hypergraph construction module and prediction module. We illustrate the sequential user-item interactions in different timestamps from short-term and long-term granularity. The figure also shows that the target user has a pairwise relation with one item, while the item has multi-modal features such as visual, acoustic and textual. A user might have different tastes on modalities of an item, for example, a user is attracted by the frames, but might turn out to be disappointed with its poor sound tracks. Multiple modalities have varying contributions to user preferences. Each item can be treated as most current interactions from target user and the time-aware selection windows capture a time slot user behavior interacting on various items. All the short and long-term user intent and item embedding are fed into attention layer to represent each target user preference.

From group-level aspect, most item own more than one user. We extract item information from user-items sequential historical records and generate group-aware hyperedges. We can see in Figure 3 that there are three different colored areas. Every area denotes a hyperedge and a group of users connected by one unimodal feature in each hyperedge. We call this hyperedge Interest-based user hyperedge, and our task is to learn a user-interest matrix, leading to construct these hyperedges. Each hypergraph in the figure represents a group of users interacting with same item in the current time altogether and have different tendencies. We can then easily learn the group-aware information to enhance individual’s representation. Besides, we have the opportunity to infer the preference of each user to make our prediction more accurate.

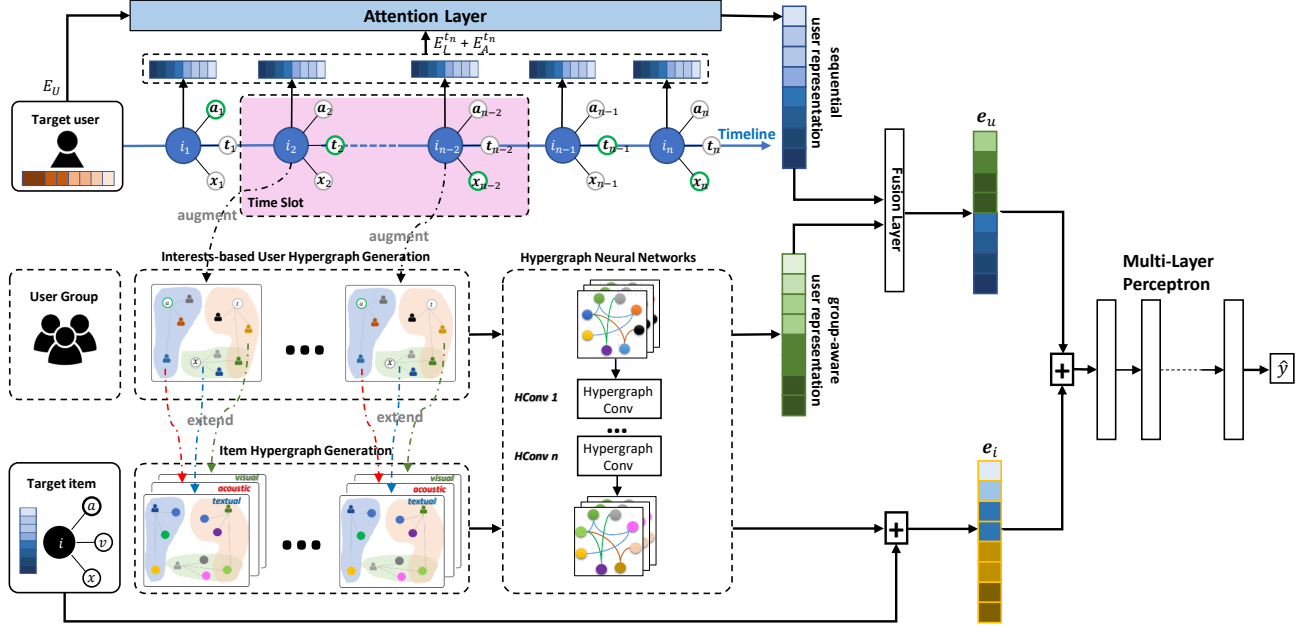
According to the group-level hyperedges, we can naturally find that each item can map to several users, while each user also has multiple interactions with various items. Here we cluster item information to build item hyperedges. There are several layers for each modality which extends from interests-based user hyperedges. The generation model will then go through the whole time period. We can now easily capture each higher-order structural relationship among items and enrich the representation of each items.

We leverage hypergraph convolutional operators to learn rich representation capturing local and higher-order structural relationships [9]. In the prediction module, we fuse group-aware user representation and sequential user representation. We then feed into a multi-layer perceptron and output the click-through rate prediction.

**2.2.1 Temporal User behavior Attention Module.** One user’s historical interaction with items can span multiple times. A straightforward way is to apply RNN-type methods to analyze the sequence  $\mathcal{S}(u, i_1, i_2, i_3, \dots)$ . However, these models fail to capture both short-term and long-term dependencies. We thus perform a sequential analysis using the proposed temporal user behavior attention mechanism.

**Embedding Layer** As depicted in Figure 3, the long-term user interaction can be represented by all the items the user has interacted with in a certain time slot  $t_n$ . In the user embedding mapping stage, to depict user behaviour features, we use their metadata and profiles and define an embedding matrix  $\mathbf{E}_U \in \mathbb{R}^{|T| \times d}$  for each user  $u_j$ . We also maintain an item embedding matrix  $\mathbf{M}_I \in \mathbb{R}^{|I| \times d}$  and a multi-modal attribute embedding matrix  $\mathbf{M}_A \in \mathbb{R}^{|A| \times d}$ . The two matrices project the high-dimensional one-hot representation of an item or multi-modal attribute to low-dimensional dense representations. Given a  $l$ -length time granularity sequence, we apply a time-aware slot window to form the input item embedding matrix  $\mathbf{E}_I^{t_n} \in \mathbb{R}^{l \times d}$ . Besides, we also form an embedding matrix  $\mathbf{E}_A^{t_n} \in \mathbb{R}^{k \times d}$  for each item from the entire multi-modality attribute embedding matrix  $\mathbf{M}_A$ , where  $k$  is the number of item modalities. The sequence representation  $\mathbf{E}^{t_n} \in \mathbb{R}^{n \times d}$  can be obtained by summing three embedding matrices:  $\mathbf{E}^{t_n} = \mathbf{E}_U + \mathbf{E}_I^{t_n} + \mathbf{E}_A^{t_n}$ .

**Attention Layer** We develop the sequential user behavior encoder by utilizing attention mechanism. We proposed to use self-attention layer, i.e., transformer which has also been applied in time series prediction [27]. In contrast to CNN, RNN-based approaches and Markov Chains-based models [14], we adopt self-attention as the basic model to capture the temporal pattern in user-items interaction sequence. A self-attention module generally consists of two



**Figure 3: The structure of HyperCTR: two views of hypergraphs are constructed based on user-item correlations at different time slot and the Hypergraph Neural Networks is able to capture the correlations in multi-hop connections. The attention layer can capture dynamic pattern in interaction sequences. Both the group-aware and sequential user embedding fuse to represent each individual, meanwhile, the target item embedding and a set of homogeneous item-item hypergraph embeddings are considered to learn the final prediction with the multi-layer perceptron.**

sub-layers, i.e., a multi-head self-attention layer and a point-wise feed-forward network. The multi-head self-attention mechanism has been adopted for effectively extracting the information selectively from different representation subspaces [40]. The multi-head self-attention is defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (1)$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (2)$$

where the projections are parameter matrices  $W_i^Q \in \mathbb{R}^{d \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d \times d_v}$  and  $W^O \in \mathbb{R}^{hd_v \times d}$ . The attention function is implemented by scaled dot-product operation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where  $(Q = K = V) = E$  are the linear transformations of the input embedding matrix, and  $\frac{1}{\sqrt{d_k}}$  is the scale factor to avoid large values of the inner product, since the multi-head attention module is mainly build on the linear projections.

In addition to attention sub-layers, we applied a fully connected feed-forward network, denoted as  $\text{FFN}(\cdot)$ , which contains two linear transformations with a ReLU activation in between.

$$\text{FFN}(x) = \text{ReLU}(0, xW_1 + b_1)W_2 + b_2 \quad (4)$$

where  $W_1, b_1, W_2, b_2$  are trainable parameters.

**2.2.2 Hypergraph Convolution Network (HGCN).** At each time slot, we aim to exploit the correlations among users and items for their high-order rich embeddings, in which the correlated users or items can be more complex than pairwise relationship, which is difficult to be modeled by a graph structure. On the other hand, the data representation tends to be multi-modal, such as the visual, text and social connections. To achieve that, each user should connect with multiple items with various modality attributes, while each item should correlated with several users. This naturally fits the assumption of the hypergraph structure for data modeling. Compared with simple graph, on which the degree for all edges is mandatory to be 2, a hypergraph can encode high-order data correlation using its degree-free hyperedges [9]. In our work, we construct a  $\mathcal{G}(u, i)$  to present user-item interactions over different time slots. Then, we aim to distill some hyperedges to build user interest-based hypergraph  $\mathcal{G}_g^{t_n}$  and item hypergraph  $\mathcal{G}_i^{t_n}$  to aggregate high-order information from all neighborhood. We concatenate the hyperedge groups to generate the hypergraph adjacent matrix  $H$ . The hypergraph adjacent matrix  $H$  and the node feature are fed into a convolutional neural network (CNN) to get the node output representations. We build a hyperedge convolutional layer  $f(X, W, \Theta)$  as follows:

$$X^{(l+1)} = \sigma\left(D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} X^{(l)} \Theta^{(l)}\right) \quad (5)$$

where define  $X, D_v, D_e$  and  $\Theta$  is the signal of hypergraph at  $l$  layer,  $\sigma$  denotes the nonlinear activation function. The GNN model is based on the spectral convolution on the hypergraph.

**2.2.3 Prediction Module and Losses.** We want to incorporate both user sequential embeddings and group-aware high-order information for a more expressive representation of each user in the sequence. We propose the fusion layer to generate the representation of user  $u$  at  $t_n$ . Existing works on multiple embeddings use concatenation as fusion [15], resulting in suboptimal interactions. We utilize the fusion process that transforms the input representations into a heterogeneous tensor [23]. We use the user sequential embedding  $\mathbf{E}^{t_n}$  and group-aware hypergraph embedding  $\mathbf{E}_g^{t_n}$ . Each vector  $\mathbf{E}$  is augmented with an additional feature of constant value equal to 1, denoted as  $\mathbf{E} = (\mathbf{E}, 1)^T$ . The augmented matrix  $\mathbf{E}$  is projected into a multi-dimensional latent vector space by a parameter matrix  $\mathbf{W}$ , denoted as  $\mathbf{W}^T \mathbf{E}_m$ . Therefore, each possible multiple feature interaction between user and group-level is computed via outer product,  $\mathcal{T} = f(\mathbf{W}^T \cdot \tilde{\mathbf{E}}_m)$ , expressed as:

$$\mathcal{T}_U = \mathbf{W}^T \cdot (\mathbf{E}^{t_n} \otimes \mathbf{E}_g^{t_n}) \quad (6)$$

Here  $\otimes$  denotes outer product,  $\tilde{\mathbf{E}}_m$  is the input representation from user and group level. It is a two-fold heterogeneous user-aspect tensor  $\mathcal{T}_U$ , modeling all possible interrelation, i.e., user-item sequential outcome embeddings  $\mathbf{E}^{t_n}$  and group-aware aggregation features  $\mathbf{E}_g^{t_n}$ .

When predicting the CTR of user for items, we take both sequential user embedding and item embedding into consideration. We calculate the user-level probability score  $y$  to a candidate item  $i$ , to clearly show how the function  $f$  works. The final estimation for the user CTR prediction probability is calculated as:

$$\hat{y} = f(\mathbf{e}_u, \mathbf{e}_i; \Theta) \quad (7)$$

where  $\mathbf{e}_u$  and  $\mathbf{e}_i$  denote user and item-level embeddings, respectively.  $f$  is the learned function with parameter  $\Theta$  and implemented as a multi-layer deep network with three layers, whose widths are denoted as  $\{D_1, D_2, \dots, D_N\}$  respectively. The first and second layer use *ReLU* as activation function while the last layer uses sigmoid function as  $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$ . As for the loss function, we take an widely used end-to-end training approach, Cross Entropy Loss[8, 28, 39], and it is formulated as:

$$L(\mathbf{e}_u, \mathbf{e}_i) = y \log \sigma(f(\mathbf{e}_u, \mathbf{e}_i)) + (1 - y) \log(1 - \sigma(f(\mathbf{e}_u, \mathbf{e}_i))) \quad (8)$$

where  $y \in \{0, 1\}$  is ground-truth that indicates whether the user clicks the micro-video or not, and  $f$  represents the multi-layer deep network.

## 2.3 Hypergraph Generation Modules

We aim to distill the user-level hypergraph group to enhance the representations of input data. We adopt a pre-training way to learn user group latent preference correlation to different modalities from items. However, as model trained is prone to suffer from unlabelled data problem, there is no explicit information to associate user and each item's modality. We further incorporate additional self-supervised signals with mutual information to learn the intrinsic data correlation [20, 40].

**2.3.1 Interest-based User Hypergraph Generation Modeling.** We aim to utilize self-supervised learning for the user-interest matrix  $\mathbf{F} \in \mathbb{R}^{L \times d}$ , where  $L$  denote the user counts and  $d$  denote the number of multi-modalities according to items. We trained the weights  $\{\theta_a, \theta_b, \theta_c\}$  for each modalities. We define  $\{\alpha, \beta, \gamma\}$  to denote the

degree of interest of each modalities from the item features. A threshold  $\delta$  was applied to measure which modality contributes the most for user-item interaction. We first maximize the mutual information between users  $u$  and item's multi-modal attributes  $M_{i_n}^{t_n}$ . For each user and item, the metadata and attributes provide fine-grained information about them. We aim to fuse user and multimodal-level information through modeling user-multimodal correlation. It is thus expected to inject useful multi-modal information into user group representation. Given an item  $i$  and the multi-modal attributes embedding matrix  $M_{i_n}^{t_n} \in \mathbb{R}^{|\mathcal{A}| \times d}$ , we treat user, item and its associated attributes as three different views denoted as  $\mathbf{E}_U, \mathbf{E}_I^{t_n}$  and  $\mathbf{E}_A^{t_n}$ . Each  $\mathbf{E}_A^{t_n}$  is associated with a embedding matrix  $M_k \in M_{i_n}^{t_n} = \{v_{i_n}^{t_n}, a_{i_n}^{t_n}, x_{i_n}^{t_n}\}$ . We design a loss function by the contrastive learning framework that maximizes the mutual information between the three views. Following Eq 8, we minimize the User Interest Prediction (UIP) loss by:

$$L_{UIP}(u, i, \mathbf{E}_{A_i}) = \mathbb{E}_{a_j \in \mathbf{E}_{A_i}} \left[ f(u, i, a_j) - \log \sum_{\tilde{a} \in \mathbf{E}_{A_i} \setminus \mathbf{E}_{A_i}} \exp(f(u, i, \tilde{a})) \right] \quad (9)$$

where we sample negative attributes  $\tilde{a}$  that enhance the association among user, item and the ground-truth multi-modal attributes, " $\setminus$ " defines set subtraction operation. The function  $f(\cdot, \cdot, \cdot)$  is implemented with a simple bilinear network:

$$f(u, i, a_j) = \sigma \left[ \left( \mathbf{E}_I^T \cdot \mathbf{W}_{UIP} \cdot \mathbf{E}_{A_j} \right) \cdot \mathbf{E}_U \right] \quad (10)$$

where  $\mathbf{W}_{UIP} \in \mathbb{R}^{d \times d}$  is a parameter matrix to learn and  $\sigma(\cdot)$  is the sigmoid function. We define the loss function  $L_{UIP}$  for a single user, which will can be extended over the user set in a straightforward way. The outcome from  $f(\cdot)$  for each user can be constructed as a user-interest matrix  $\mathbf{F}$  and compared with the threshold  $\delta$  to output the  $L$ -dimensions vector  $\mathbf{v} \in \mathbb{R}^{1 \times L}$ .

**2.3.2 Item Hypergraph Construction.** We exploit how to transform a sequential user-item interactions into a set of homogeneous item-level hypergraph. We construct a set of homogeneous hypergraphs  $\mathcal{G}_I$ , from node sets  $I$  as follow:

$$\mathcal{G}_I = \{\mathcal{G}_{I,\text{group}}, \mathcal{G}_{I,1}, \dots, \mathcal{G}_{I,Q}\} \quad (11)$$

where  $\mathcal{G}_{I,j} = \{I, \mathcal{E}_{I,j}\}$ , and  $\mathcal{E}_{I,j}$  denote hyperedges in  $\mathcal{G}_{I,j}$ . Note that all the homogeneous hypergraphs in  $\mathcal{G}_I$  share the same node set  $I$ . For a node  $i \in I$ , a hyperedge introduced in  $\mathcal{E}_{I,j}$  of  $\mathcal{G}_{I,j}$ , which connects to  $\{i | i \in I, (u, i) \in \mathcal{E}_{T_n}\}$ , i.e., the vertices in  $I$  that are directly connected to  $u$  by  $\mathcal{E}_{T_n}$  in time period  $T_n$ . According to Figure 3, in the user-item sequential interaction network, the user  $u$  clicks three items  $v$ , which corresponds to a hyperedge that connects these three items in the homogeneous hypergraph  $\mathcal{G}_I$ . The special homogeneous hypergraph  $\mathcal{G}_{I,\text{group}} \in \mathcal{G}_I$  are defined as  $G \left( I, \bigcup_{j=1}^k \mathcal{E}_{I,j} \right)$ . Note that the cardinalities of hyperedge sets in the constructed hypergraph are:  $|\mathcal{E}_{I,j}| \leq |U|$  and  $|\mathcal{E}_{I,\text{group}}| \leq k|U|$  for  $j \leq k$ . The total number of hyperedges in the homo-hypergraph is proportional to the number of nodes and edge types in the input sequence:  $O(k(|I| + |V|))$ . Thus, the transformation easily scales to large inputs.

**2.3.3 Information Augmentation.** The increasing data sparsity problem is one of our main motivations in tackling with CTR prediction task. To address the interaction sparsity problem, some information augmentation methods have been proposed [21, 35], however,

**Table 1: Statistics of the dataset. (v, a and t denote the dimensions of visual, acoustic, and textual modalities, respectively.)**

Dataset	#Items	#Users	#Interactions	Sparsity	v.	a.	t.
Kuaishou	3,239,534	10,000	13,661,383	99.98%	2048	-	128
MV1.7M	1,704,880	10,986	12,737,619	-	128	128	128
MovieLens	10,681	71,567	10,000,054	99.63%	2048	128	100

they only consider in the case of single modality and cannot handle the scenarios with multi-modal features. We propose two data augmentation strategies, which use user behavior information and item multi-modal information to learn the subgraph embedding. We transform the initial user-item heterogeneous hypergraph into two homogeneous hypergraphs from the perspective of users and items respectively.

**User Behavior Information Augment Strategy** We have utilized temporal user interaction logs to represent user-level embedding. However, the heterogeneous nature between users and items aggravates the difficulty in network information fusion. A common observation is that the user usually interacts with only a small number of items while an item can only be exposed to a small number of users, which results in a sparse user-item network and limits the effectiveness of embedding representation. To mitigate the issue, we utilize the self-supervised user interest matrix  $F$  to build the user-user homogeneous graphs, which contains multiple hyperedges, and is regarded as hypergraph. It is denoted as  $\mathcal{G}_g^{t_n}$  mentioned in Definition 2.

**Item Multi-modal Information Augment Strategy** It is a common observation that if two users both link to the same modality of items, then they have some common interest [37]. We are thus motivated to add an edge between them in  $\mathcal{G}_g^{t_n}$ . Similarly, if some items link to the same set of users, they share the same target user group. We thus add an hyperedge between them in  $\mathcal{G}_i^{t_n}$ .

According to the two information augmentation strategies, we transform the first-order neighbor relations of user-item to second-order neighbor relations of user-user and item-item, and represent the complex relationship as a multiple hypergraph structure. Compared with single hop neighbors, in our case nodes have more hop neighbors, which can be used to alleviate the problem of graph sparsity. The items in each hyperedge in  $\mathcal{G}_i^{t_n}$  maintain some intrinsic attribute correlation due to which they connect with the same user preference. Adding edge information while aggregating information from neighbor nodes can exchange heterogeneous topology information between  $\mathcal{G}_g^{t_n}$  and  $\mathcal{G}_i^{t_n}$ . The information fusion processes on the two graphs are interdependent.

## 3 EXPERIMENTS AND RESULTS

### 3.1 Experimental Settings

**3.1.1 Datasets.** Existing CTR prediction models mostly utilize unimodal datasets [18, 20, 26, 32]. In contrast, we introduce multiple modalities into CTR prediction. As mentioned above, micro-video datasets contain rich multimedia information and include multiple modalities - visual, acoustic and textual. We experimented with three publicly available datasets: Kuaishou, MV1.7M and MovieLens 10M which are summarized in Table 1.

**Kuaishou:** This dataset is released by the Kuaishou [17]. There are multiple interactions between users and micro-videos. Each behaviour is also associated with a timestamp, which records when the event happens. The timestamp has been processed to modify the absolute time, but the sequential temporal order is preserved w.r.t to the timestamp.

**Micro-Video 1.7M:** This dataset was proposed in [4]. The interaction types include “click” and “unclick”. Each micro-video is represented by a 128-dimensional visual embedding vector of its thumbnail. Each user’s historical interactions are sorted in chronological order.

**MovieLens:** The MovieLens dataset is obtained from the MovieLens 10M Data<sup>3</sup>. We assume that a user has an interaction with a movie if the user gives it a rating of 4 or 5. We use the pre-trained ResNet[11] models to obtain the visual features from key frames extracted from micro-video. For acoustic modality, we separate audio tracks with FFmpeg6 and adopt VGGish [12] to learn the acoustic deep learning features. For textual modality, we use Sentence2Vector [22] to derive the textual features from micro-videos’ descriptions.

**3.1.2 Baseline Models.** We compare our model with strong baselines from both sequential CTR prediction and recommendation. Our comparative methods are: 1) **GRU4Rec** [13] based on RNN. 2) **THACIL** [4] is a personalized micro-video recommendation method for modeling user’s historical behaviors, which leverages category-level and item-level attention mechanisms to model the diverse and fine-grained interests respectively. It adopts forward multi-head self-attention to capture the long-term correlation within user behaviors. 3) **DSTN** [24] learns the interactions between each type of auxiliary data and the target ad, to emphasize more important hidden information, and fuses heterogeneous data in a unified framework. 4) **MIMN** [25] is a novel memory-based multi-channel user interest memory network to capture user interests from long sequential behavior data. 5) **ALPINE** [17] is a personalized micro-video recommendation method which learns the diverse and dynamic interest, multi-level interest, and true negative samples. It utilizes a temporal graph-based LSTM network to model users’ dynamic and diverse interests from click sequence, and capture uninterested information from the true negative sample. It introduces a user matrix to enhance user interest modeling by incorporating multiple types of interactions. 6) **AutoFIS** [19] automatically selects important 2<sup>nd</sup> and 3<sup>rd</sup> order feature interactions. The proposed methods are generally applicable to many factorization models and the selected important interactions can be transferred to other deep learning models for CTR prediction. 7) **UBR4CTR** [26] has a retrieval module and it generates a query to search from the whole user behaviors archive to retrieve the most useful behavioral data for prediction. The retrieved data is then used by an attention-based deep network to make the final prediction.

**3.1.3 Parameter Settings.** We randomly split all datasets into training, validation, and testing sets with 7:2:1 ratio, and create the training triples based on random negative sampling. For testing set, we pair each observed user-item pair with 1000 unobserved micro-videos that the user has not interacted with before.

<sup>3</sup><http://files.grouplens.org/datasets/movielens/>



**Table 2: Parameter Settings**

Methods	#Batch size	#Dropout	#Learning rate
GRU4Rec	200	0.1	0.05
THACIL	128	0.2	0.001
DSTN	128	0.5	0.001
MIMN	200	0.2	0.001
ALPINE	2048	0.3	0.001
AutoFIS	2000	0.6	0.005
UBR4CTR	200	0.5	0.001

For our baseline methods, we use the implementation and settings provided in their respective papers. More details show as follow items and Table 2.

- **GRU4Rec** We applies GRU to model user click sequence for reproduce this model. We represent the items using embedding vectors rather than one-hot vectors.
- **THACIL** The number of micro-videos per user is set to 160. The temporal block size is set to 20. For users having more items than 160, we just preserve as much as 160 items. For users having less items, we pad all-zero vectors to augment.
- **DSTN** We set the dimension of the embedding vectors for each feature as 10, because the number of distinct features is huge. We set the number of fully connected layers in DSTN is 2, each with dimensions 512 and 256.
- **MIMN** Layers of FCN (fully connected network) are set by  $200 \times 80 \times 2$ . The number of embedding dimension is set to be 16, which is the same as the dimension of memory slots. The number of hidden dimension for GRU in MIU is set to be 32. Number of memory slots in both NTM and MIU is a parameter that is examined carefully in the ablation study section. We take AUC as the metric for measurement of model performance.
- **ALPINE** We utilized the 64-d visual embedding to represent the micro-video. The length of users’ historical sequence is set to 300. If it exceeds 300, we truncated it to 300; otherwise, we padded it to 300 and masked the padding in the network.
- **AutoFIS** We implement the two-stage algorithm AutoFIS to automatically select important low-order and high-order feature interactions with FM-based model.
- **UBR4CTR** The datasets are processed into the format of comma separated features. A line containing user, item and context features is treated as a behavior document.

In HyperCTR and all its variants use Adam optimizer. For training, we randomly initialize model parameters with a Gaussian distribution and use the ReLU as the activation function. We then optimized the model with stochastic gradient descent (SGD). We search the batch size in 128, 256, 512, the the latent feature dimension in 32, 64, 128, the learning rate in 0.0001, 0.0005, 0.001, 0.005, 0.01 and the regularizer in 0, 0.00001, 0.0001, 0.001, 0.01, 0.1. As the findings are consistent across the dimensions of latent vectors, we have shown the result of 64, a relatively large number that returns good performance whose details can be found sensitivity analysis.

**3.1.4 Evaluation Metrics.** We evaluate the CTR prediction performance using two widely used metrics. The first one is Area Under

**Table 3: The overall performance of different models on Kuaishou, Micro-Video 1.7M and MovieLens datasets in %.**

Method	Kuaishou		MV1.7M		MovieLens	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
GRU4Rec	0.7367	0.5852	0.7522	0.6613	0.7486	0.6991
THACIL	0.6640	0.5793	0.6842	0.6572	0.6720	0.6791
DSTN	0.7722	0.5672	0.7956	0.6492	0.8008	0.6162
MIMN	0.7593	0.5912	0.7486	0.6862	0.7522	0.6751
ALPINE	0.6840	0.5632	0.7130	0.6591	0.7390	0.6163
AutoFIS	0.7870	0.5756	0.8010	0.5404	0.7983	0.5436
UBR4CTR	0.7520	0.5710	0.8070	0.5605	0.8050	0.5663
<b>HYPERCTR</b>	<b>0.8120</b>	<b>0.5548</b>	<b>0.8670</b>	<b>0.5160</b>	<b>0.8360</b>	<b>0.5380</b>
Improv.(%)	3.18%	1.49%	7.43%	4.51%	3.85%	1.03%

ROC curve (AUC) which reflects the pairwise ranking performance between click and non-click samples. The other metric is log loss. Log loss is to measure the overall likelihood of the test data and has been widely used for the classification tasks [29, 30].

### 3.2 Quantitative Performance Comparison

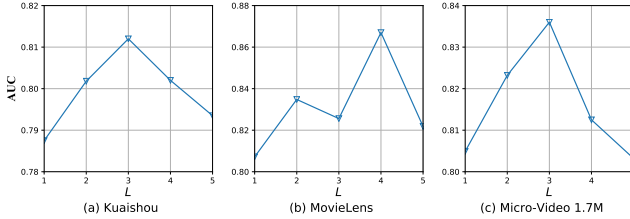
Table 3 presents the AUC score and Logloss values for all models. When different modalities re used, all models show an improved performance when the same set of modalities containing visual, acoustic and textual features are used in MV1.7M and MoiveLens(10M). We also note that: (a) the performance of our model has improved significantly compared to the best performing baselines. AUC is improved by 3.18%, 7.43% and 3.85% on three datasets, respectively, and Logloss is improved by 1.49%, 4.51% and 1.03%, respectively; and (b) the improvement in our model demonstrates that the uni-modal features do not embed enough temporal information which the baselines cannot exploit effectively. The baseline methods cannot perform well if the patterns that they try to capture do not contain multi-modal features in the user-item interaction sequence.

### 3.3 HyperCTR Component Analysis

**3.3.1 Role of Multimodality .** To explore the effect of different modalities, we compare the results on different modalities on the three datasets, as shown in Table 4. We make the following observations: 1) Our main method outperforms those with single-modal features on three datasets. It demonstrates that representing users with multi-modal information achieves a better performance. It also demonstrates that the construction of hyperedges can capture user’s modal-specific preference from graph information. 2) The visual-modal is the most effective one among three modalities. It can be naturally understood that if a user clicks what to watch, one usually pays more attention to the visual information than other modality. 3) The acoustic-modal shows more important information for user click compared with the textual features. This is expected as the background music is more attractive to users. 4) Textual modality contributes least towards click-through rate prediction. This is natural since the description of text always has low quality, and contains noisy, special symbols and even irrelevant to the micro-video content in Kuaishou and MV1.7M datasets. However, in MovieLens data corpus, this modality has smaller gap with the other modalities. This is because the text in MovieLens is highly related to the content. 4) Compared with GCN, our proposed model achieved better performance in all datasets. As shown in Table 4, based on Kuaishou datasets, when only two features are used for graph/hypergraph, our

**Table 4: Performance in terms of AUC & Logloss w.r.t different modalities on the three datasets in %.**

Method	Kuaishou		MV1.7M		MovieLens	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
multi-modal	<b>0.8120</b>	<b>0.5548</b>	<b>0.8670</b>	<b>0.5160</b>	<b>0.8360</b>	<b>0.5380</b>
visual-modal	0.8110	0.5560	0.8567	0.5167	0.8259	0.5376
acoustic-modal	-	-	0.8260	0.5171	0.8134	0.5373
textual-modal	0.7720	0.5756	0.8158	0.5175	0.8123	0.5364
(-) hypergraph	0.8034	0.5554	0.8137	0.5426	0.8064	0.5673

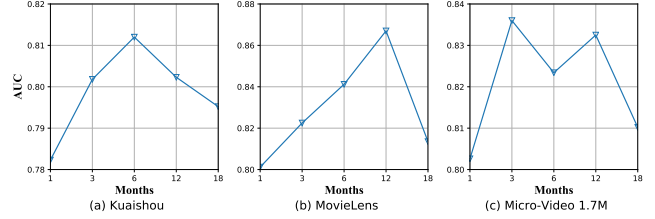


**Figure 4: Performance comparison with different number of HGCN layers under AUC**

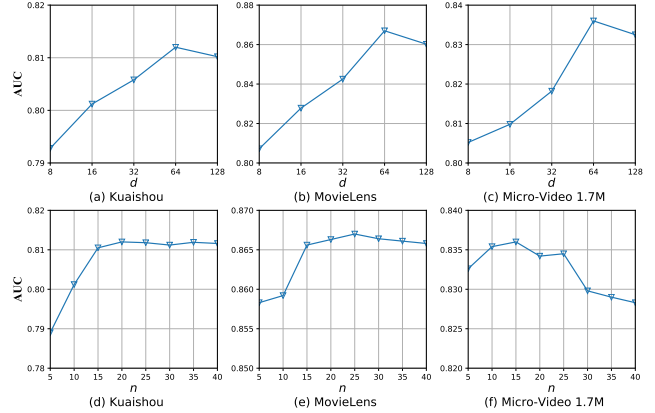
model can still obtain slight improvement. With more features in the other two datasets, our model achieves much better performance compared with GCN. This phenomenon is consistent with our argument that when multi-modal features are available, hypergraph has the advantage of combining such multi-modal information in the same structure by its flexible hyperedges.

**3.3.2 Role of HGCN Layers.** To explore how the high-order connections in the hypergraph can help to uncover hidden item correlations and thus contribute to the final prediction. We compare the performance of HyperCTR by varying the number of hypergraph convolutional layers. As shown in Figure 4, when we apply only one convolution layer for our sequential model, each node embedding aggregates only information from others connected with them directly by the hyperedge. Our model performs poorly in all three datasets. By stacking three HGCN layers, it can bring in significant improvement compared with a model with just one convolution layer. We can infer that hyper-graph and HGCN are useful options for extracting expressive item semantics and it is important to take the high-order neighboring information in hypergraph into consideration. On Kuaishou and MV1.7M, since the data is very sparse, it is not necessary to further increase the number of convolutional layers. Three HGCN layers are enough for extracting the user- and item-level semantics at different time slots. On MovieLens, more convolutional layers can further improve the embedding process. This demonstrates the effectiveness of hypergraph and HGCN in modeling the temporal user and item correlations.

**3.3.3 Role of Time Granularity.** An important parameter which can effect the performance of HyperCTR is the granularity of the time slot. According to Figure 5, we show the performance of the proposed model by varying the granularity from 1 month to 18 months. When the granularity is small, we find that the model cannot achieve the best performance since the interactions are extremely sparse and not sufficient for building up a set of expressive user and item embeddings. While enlarging the granularity, we find that the performance of HyperCTR is increasing in all the datasets. In



**Figure 5: Performance comparison with various time granularity under AUC**



**Figure 6: Impact of embedding dimension (top row) and sampled neighbor size (bottom row)**

Kuaishou datasets, it reaches the best performance when the time granularity is set to half a year. However, for MovieLens, the optimized granularity is almost one year since the item in MovieLens is movie, its propagation speed is relatively slow and the impact time is relatively long. In MV1.7M datasets, the optimized granularity is around three months, which is smaller than that for the other datasets since the micro-video sharing platform attracts more interactions for each time slot for the temporal user preference representations. If we further enlarge the granularity, the performance will decrease since it underestimates the change of user preference and may introduce noise to the model.

### 3.4 HyperCTR Model Parameter Study

**3.4.1 Hyperparameter Sensitivity Analysis.** We study sensitivity of HyperCTR on the key hyperparameters using the three public datasets. The hyper-parameters play important roles in HGCN-based model, as they determine how the node embeddings are generated. We conduct experiments to analyze the impact of two key parameters which are the embedding dimension  $d$  and the size of sampled neighbors set for each node. According to Figure 6, we can note that: 1) When  $d$  varies from 8 to 256, all evaluation metrics increase in general since better representations can be learned. However, the performance becomes stable or slightly worse when  $d$  further increases. This may be due to over-fitting. 2) When the neighbor size varies from 5 to 40, all evaluation metrics increase at first as suitable amount of neighborhood information are considered. When the size of neighbors exceeds a certain value, performance decreases slowly which may be due to irrelevant neighbors. The most ideal neighbor size is in the range of 15 to 25.



**3.4.2 Scalability Analysis.** As GCN-based networks are complex and contain such a large number of nodes in the real world application scenario, it is necessary for a model being feasible to be applied in the large-scale datasets. We investigate the scalability of HyperCTR model optimized by gradient descent, which deploys multiple threads for parallel model optimization. Our experiments are conducted in a computer server with 24 cores and 512GiB memory. We run experiments with different threads from 1 to 24. We depict in Figure 7 the speedup ratio vs. the number of threads. The speedup ratio is very close to linear, which indicates that the optimization algorithm of the HyperCTR is reasonably scalable.

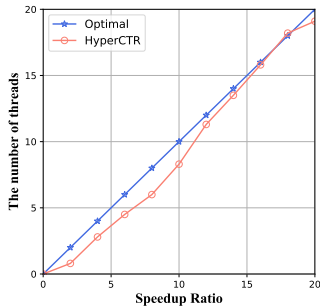


Figure 7: Scalability of HyperCTR

**3.4.3 Model Training.** To depict our model training process, we plot the learning curves of HyperCTR, as shown in Figure 8. The three subfigures are the AUC curves of the multi-modal hypergraph framework when training on three datasets. Every epoch of the  $x$ -axis is corresponding to the iteration over 5% of the training set.

## 4 RELATED WORK

*CTR prediction.* Learning the effect of feature interactions seems to be crucial for accurate CTR prediction. Factorization Machines (FMs) [2, 31] are proposed to model pairwise feature interactions in terms of the vectors corresponding to the involved features. AutoFIS [19] and UBR4CTR [26] further improve FM by removing the redundant feature interactions and retrieving a limited number of historic behavior that are most useful for each CTR prediction target. However, a FM-based model considers learning shallow representation, and it thus is unable to model the features faithfully. Deep Neural Networks (DNNs) are exploited for CTR prediction in order to automatically learn feature representations and higher-order feature interactions. DSTN [24] integrates heterogeneous auxiliary data (i.e., contextual, clicked and unclicked ads) in a unified framework

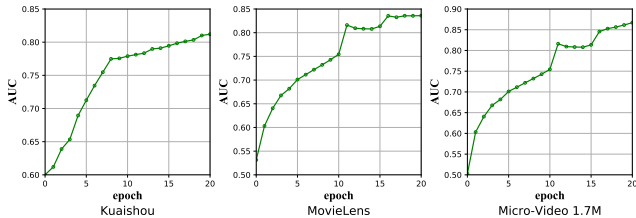


Figure 8: Learning process of HyperCTR.

based on the DNN model. Further, the other stream of models focus more on mining temporal patterns from sequential user behavior. GRU4Rec [13] is based on RNN. It is the first work which uses the recurrent cell to model sequential user behavior. MIMN [25] applies the LSTM/GRU operations for modeling users' lifelong sequential behavior.

*Exploiting multi-modal representation.* Some works focus on the multi-modal representation in the area of multi-modal CTR prediction. Existing multi-modal representations have mostly been applied to recommender systems and have been grouped into two categories: joint representations and coordinated representations [34]. Joint representations usually combine the uni-modal information and project into the same representation space [5, 6, 38]. Although, visual or textual data and are increasingly used in the multi-modal domain [16], they are suited for situations where all of the modalities are present during inference, which is hardly guaranteed in social platforms. Different from the joint representations, the coordinated models learn separate representations for each modality but coordinate them with constraints [34]. Since the modal-specific information is the factor for the differences in each modality signals, the model-specific features are inevitably discarded via similar constrains. In contrast, we introduce a novel model which respectively models the information augmentation and group-aware network problems to address the limitations in existing works.

*Graph Convolution Network.* Our proposed model uses the GCNs technique to represent the users and items, which has been popularly used for modeling the social media data. In [10] the authors proposed a general inductive framework which leverages the content information to generate node representation for unseen data. In [36] the authors developed a large-scale deep recommendation engine on Pinterest for image recommendation. In their model, graph convolutions and random walks are combined to generate the representations of nodes. In [1] the authors proposed a graph auto-encoder framework based on message passing on the bipartite interaction graph. However, these methods cannot model the multi-modal data including cases where data correlation modeling is not straightforward [9].

## 5 CONCLUSION

In this paper, we model temporal user preferences and multi-modal item attributes to enhance the accuracy of CTR prediction. We design a novel HGCN-based framework, named HyperCTR, to leverage information interaction between users and micro-videos by considering different modalities. We also refine user presentation from two aspects: time-aware and group-aware. With the stacking of hypergraph convolution networks, a self-attention and the fusion layer, our proposed model provides more accurate modeling of user preferences, leading to improved performance.

In the future, we will continue to explore how special time point influence the CTR prediction results. We also would like to incorporate some cross-platform information (e.g., e-commerce transaction data) into our system towards better performance.

## REFERENCES

- [1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
- [2] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. 2016. Higher-order factorization machines. *arXiv preprint arXiv:1607.07195* (2016).
- [3] Alain Bretto. 2013. Hypergraph theory. *Springer* (2013).
- [4] Xusong Chen, Dong Liu, Zheng Jun Zha, Wengang Zhou, and Yan Li. 2018. Temporal Hierarchical Attention at Category- and Item-Level for Micro-Video Click-Through Prediction. *SIGIR*.
- [5] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable recommendation by leveraging reviews and images. *TOIS* 37, 2 (2019), 1–28.
- [6] Zhiyong Cheng, Shen Jialie, and Steven CH Hoi. 2016. On effective personalized music retrieval by exploring online user behaviors. In *SIGIR*. 125–134.
- [7] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *SIGIR*. 515–524.
- [8] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *IJCAI* 2019-August (2019), 2301–2307. arXiv:1905.06482
- [9] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. *AAAI* (2019), 3558–3565. arXiv:1809.09401
- [10] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. (2015). arXiv:1512.03385 [cs.CV]
- [12] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. 2017. CNN Architectures for Large-Scale Audio Classification. arXiv:1609.09430 [cs.SD]
- [13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. (2016).
- [14] Wang Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. *ICDM* 2018-November (2018), 197–206. arXiv:1808.09781
- [15] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. Spam review detection with graph convolutional networks. (2019), 2703–2711.
- [16] Xiang Li, Chao Wang, Jiwei Tan, Xiaoyi Zeng, Dan Ou, and Bo Zheng. 2020. Adversarial Multimodal Representation Learning for Click-Through Rate Prediction. *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020* (2020).
- [17] Yongqi Li, Chaoran Cui, Meng Liu, Xin Shun Xu, Jianhua Yin, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. *MM* (2019), 1464–1472.
- [18] Bin Liu, Jinkai Yu, Ruiming Tang, Huifeng Guo, Yingzhi Chen, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. *WWW* (2019), 1119–1129. arXiv:1904.04447
- [19] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiquang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. *KDD* (2020), 2636–2645. arXiv:2003.11235
- [20] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised Learning: Generative or Contrastive. (2020), 1–23.
- [21] Zemin Liu, Vincent W. Zheng, Zhou Zhao, Hongxia Yang, and Jing Ying. 2018. Subgraph-augmented Path Embedding for Semantic User Search on Heterogeneous Social Network. (2018).
- [22] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. (2018). arXiv:1803.02893 [cs.CL]
- [23] Sijie Mai, Haifeng Hu, and Songlong Xing. 2020. Modality to Modality Translation: An Adversarial Representation Learning and Graph Fusion Network for Multimodal Fusion. In *AAAI*, Vol. 34, 164–172.
- [24] Wentao Ouyang, Xiuwu Zhang, Li Li, Heng Zou, Xin Xing, Zhaojie Liu, and Yanlong Du. 2019. Deep Spatio-Temporal Neural Networks for Click-Through Rate Prediction. In *SIGKDD*. 2078–2086.
- [25] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. *KDD* (2019), 2671–2679. arXiv:arXiv:1905.09248v3
- [26] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User Behavior Retrieval for Click-Through Rate Prediction. *SIGIR* (2020), 2347–2356. arXiv:2005.14171
- [27] Jack W Rae, L O N G Ange, and Chloe Hillier. 2019. Compressive TRansformers. (2019), 1–19. arXiv:arXiv:1911.05507v1
- [28] Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, and Jun Wang. 2018. Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising. *TKDE* 30, 99 (2018), 645–659.
- [29] Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. 2018. Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising. *TKDE* 30, 4 (2018), 645–659. arXiv:1803.02194
- [30] Kan Ren, Weinan Zhang, Yifei Rong, Haifeng Zhang, Yong Yu, and Jun Wang. 2016. User response learning for directly optimizing campaign performance in display advertising. In *CIKM*. 679–688.
- [31] Steffen Rendle. 2010. Factorization machines. In *ICDM*. IEEE, 995–1000.
- [32] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards Automated Neural Interaction Discovery for Click-Through Rate Prediction. *KDD* (2020), 945–955. arXiv:2007.06434
- [33] Jiayi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*.
- [34] 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 *ICM*. 1437–1445.
- [35] Jixing Xu, Zhenlong Zhu, Jianxin Zhao, Xuanye Liu, and Jiecheng Guo. 2020. Gemini: A Novel and Universal Heterogeneous Graph Information Fusing Framework for Online Recommendations. (2020).
- [36] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*. 974–983.
- [37] Jianglong Zhang, Liqiang Nie, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat Seng Chua. 2016. Shorter-is-Better: Venue Category Estimation from Micro-Video. In *MM*. 1415–1424.
- [38] Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. 2019. Neural multimodal belief tracker with adaptive attention for dialogue systems. In *WWW*. 2401–2412.
- [39] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep Interest Evolution Network for Click-Through Rate Prediction. (2018). arXiv:1809.03672 [stat.ML]
- [40] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S<sup>3</sup>-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. (2020), 1893–1902. arXiv:2008.07873