LightGT: A Light Graph Transformer for Multimedia Recommendation

Yinwei Wei
National University of Singapore
weyinwei@hotmail.com

Wenqi Liu
Shandong University
liuwq_bit@outlook.com

Fan Liu
National University of Singapore
liufancs@gmail.com

Xiang Wang[§]
University of Science and Technology of China
xiangwang@u.nus.edu

Liqiang Nie
Harbin Institute of Technology (Shenzhen)
nieliqiang@gmail.com

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Multimedia recommendation methods aim to discover the user preference on the multi-modal information to enhance the collaborative filtering (CF) based recommender system. Nevertheless, they seldom consider the impact of feature extraction on the user preference modeling and prediction of the user-item interaction, as the extracted features contain excessive information irrelevant to the recommendation.

To capture the informative features from the extracted ones, we resort to Transformer model to establish the correlation between the items historically interacted by the same user. Considering its challenges in effectiveness and efficiency, we propose a novel Transformer-based recommendation model, termed as Light Graph Transformer model (LightGT). Therein, we develop a modal-specific embedding and a layer-wise position encoder for the effective similarity measurement, and present a light self-attention block to improve the efficiency of self-attention scoring. Based on these designs, we can effectively and efficiently learn the user preference from the off-the-shelf items' features to predict the user-item interactions. Conducting extensive experiments on Movielens, Tiktok and Kwai datasets, we demonstrate that LigthGT significantly outperforms the state-of-the-art baselines with less time. Our code is publicly available at: https://github.com/Liuwq-bit/LightGT.

CCS CONCEPTS

• **Information systems → Recommender systems.**

KEYWORDS

Multimedia Recommendation, Transformer, Graph Convolutional Network, Recommender System

ACM Reference Format:
Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang[§], Liqiang Nie, and Tat-Seng Chua. 2023. LightGT: A Light Graph Transformer for Multimedia 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/10.1145/3539618.3591716

1 INTRODUCTION

Multimedia recommender system aims to discover the user preference for content information, in order to enrich the collaborative signal based on the collaborative filtering (CF) algorithm [21, 26, 31, 32]. Hence, it is becoming increasingly prevalent in online applications, ranging from search engines, E-commerce, to social media platforms.

To predict the user preference from the items' content information, the prior studies [3, 12, 35] mostly follow a similar pipeline: the multi-modal feature extraction, user preference modeling, and user-item similarity measurement. Although these models introduce some advanced techniques to supercharge the recommendation, such as deep neural networks [3] and graph convolutional networks (GCNs) [35], they seldom explore the extracted features, which is the foundation of the user preference modeling and subsequent interaction prediction.

Probing the existing models, we argue that the extracted features contain much information irrelevant to the recommendation task, thereby fail to sufficiently model the user preference from the item content. Taking the visual features extracted by the pre-trained ResNet [11] as an example, the discriminative signal tends to be emphasized to distinguish the images from different categories. Nevertheless, such a signal may pose an insignificant role for the user preference modeling. For instance, the features of shoelaces could help the classifier to differ the sneaker from dress shoes, whilst their importance will fall behind in the predictions of user-item interactions.

To overcome this problem, it is of great importance to distill the informative cues from off-the-shelf features for modeling user preference. The existing studies leverage the attention mechanism [3] or deep learning method [12] to distill the informative signal by explicitly or implicitly reconstructing user-item historical interactions. Nevertheless, they ignore the correlation between the items observed by the same user in the features distillation, which reveals the users' interest in the content information. This is consistent with the assumption that similar

[§]Xiang Wang is also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591716

items are prone to be recommended to the same users. That is, some features co-occurred in the items reflect the user's tastes in their content information. Therefore, establishing the correlation between the items observed by the same user is able to discover the informative features that affect the user's decision-making.

Towards this end, we resort to Transformer techniques [28] to model the correlation between the items observed by the same user. Although recent years have witnessed many successful Transformer variants in the sequential and session recommendation [25, 39, 41], it is non-trivial to directly inherit the Transformer model in our work, due to two technical challenges:

- The extracted features of items, as the inputs of Transformer models, are dependent on pre-trained extractors, and thus some irrelevant information may harm the correlation modeling of item pairs. Therefore, *how to effectively measure the affinities between the items upon the off-the-shelf features is the first challenge to conduct the Transformer-based model.*
- Unlike the computer vision and natural language processing tasks, the input tokens in recommender system will be the exponential combination of the numerous items. Hence, *facing the overload and diversity inputs, how to efficiently perform the Transformer-based model is the other challenge in deploying the model.*

To address the challenges, we propose a Light Graph Transformer model (LightGT) to establish the item-item correlation, so as to learn the superior representation of user preference from the interacted items' features. Thereinto, we develop a modal-specific embedding and a layer-wise position encoder for the effective similarity measurement, and present a light self-attention block to improve the efficiency of self-attention scoring. Specifically, we gather the user with her/his interacted items together and embed them in each modality. These embeddings help to distill the users' interested features under the supervision of historical user-item interactions. Moreover, we gain the layer-wise position encodings based on the structure information of user-item graph to enrich the tokens' representations. It can help the attention scoring by introducing the CF signal into the self-attention measurement. In the light self-attention block, we empirically remove the multi-head attention, feed-forward network (FFN), and residual connection from the vanilla model for the efficiency. Owing to these designs, we can effectively and efficiently distill the informative features and learn the user preference on content information. Ultimately, we predict the user-item interactions by combining their CF- and content-based similarities.

To justify our proposed model, we conduct extensive experiments on three datasets, including Movielens, Tiktok, and Kwai datasets. The results demonstrate that the LightGT outperforms the state-of-the-art baselines by a significant margin. And interestingly, under the same experiment setting, the light self-attention block not only boosts the model's efficiency but leads to accuracy improvements.

In a nutshell, our main contributions are three-fold:

- After probing the prior studies, we propose a new Light Graph Transformer model (LightGT) for the multimedia recommendation. To the best of our knowledge, this is the first Transformer-based model for the multimedia recommendation.
- Within our proposed LightGT model, we develop the modal-specific embedding, layer-wise position encoder, and light self-attention block, in order to efficiently and effectively optimize the user and item representations for the predictions of user-item interactions.
- Conducting extensive experiments on three datasets, we demonstrate the effectiveness and efficiency of our proposed model. The insight we observed is that the devised light Transformer structure not only decreases the time complexity but improves the performance of recommendation.

2 PRELIMINARY

In this section, we formulate the multimedia recommendation task followed by the brief introduction of vanilla Transformer model.

2.1 Multimedia Recommendation

Given a set $\mathcal{U}$ of N users and a set $\mathcal{I}$ of M items, the recommendation model targets at scoring the interests of each user to her/his unobserved items and ranking the scores in a descending order for recommendation.

Supervised by the observed user-item interactions $\mathcal{O} = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}\}$, CF-based recommendation models [26, 32] are widely used to learn the embeddings of users and items and score their similarities. Formally, we feed the user and item embeddings into a scoring function $f()$ as,

$$s_{u,i} = f(\mathbf{e}_u, \mathbf{e}_i), \tag{1}$$

where $\mathbf{e}_u \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$ denote the embeddings of user u and item i, respectively. $s_{u,i}$ is the score reflecting how likely user u will be interested in item i based on their historical behaviors. In addition, d is the dimension of the vector. By recovering the interactions in the history, the embeddings are learned during the training phases.

To enhance the CF-based recommendation models, some side information, like pictures of products and keyframes of videos, is utilized to model the user preference on the items' content. Without loss of generality, we consider the multimedia signal, involving visual, acoustic, and textual modalities, and learn the user preference in each modality. We define $m \in \mathcal{M} = \{v, a, t\}$ as the modality indicator, where v, a, and t represent the visual, acoustic, and textual modalities, respectively. Based on the multimodal information, we aim to learn the modal-specific user preference [35] and model the similarity of user-item pair under the proposed model:

$$s^m_{u,i} = g(\mathbf{f}^m_u, \mathbf{f}^m_i), \tag{2}$$

where $s^m_{u,i}$ is the content-based similarity between user u and item i. And, $\mathbf{f}^m_u \in \mathbb{R}^d$ and $\mathbf{f}^m_i \in \mathbb{R}^d$ denote the representation of u's preference and that of i on modality m, respectively. Moreover, $g()$ represents the model to be used for the similarity measurement.

Thereafter, the multimedia recommendation model is able to predict the interactions between users and items by combing the obtained CF-based and content-based similarities.

2.2 Vanilla Transformer

Recalling the vanilla Transformer, we roughly group the operations as multiple blocks with the same architecture, each of which consists of the self-attention module and position-wise FFNs. To stack these blocks, a residual connection followed by Layer Normalization is inserted between the successive blocks, facilitating the optimization of the deeper Transformer model.

Self-attention Block: By feeding a sequence of tokens (*e.g.,* words, frames, and items) into the Transformer model, the self-attention module targets at estimating the similarities between tokens and then generate the matrix of output as,

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}})\mathbf{V}, \tag{3}$$

where $softmax(\cdot)$ denotes the row-wise softmax function. We set $\mathbf{Q} = \mathbf{W}^{Q}\mathbf{H}$, $\mathbf{K} = \mathbf{W}^{K}\mathbf{H}$, and $\mathbf{V} = \mathbf{W}^{V}\mathbf{H}$ in the self-attention manner, in which $\mathbf{W}^{Q}$, $\mathbf{W}^{K}$, and $\mathbf{W}^{V}$ are the trainable weight matrices. And, $\mathbf{H}$ denotes the representations of inputted tokens at the initial block or the outputs of the previous one.

To boost the representation learning of Transformer model, the standard self-attention module is extended to the multi-head one. In particular, the tokens can be mapped into the different spaces for comprehensively measuring the similarities from different aspects, formally,

$$\begin{gathered} MultiHeadAtt(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, \cdots, head_H), \\ \text{where } head_h = Attention(\mathbf{W}_h^{Q}\mathbf{H}, \mathbf{W}_h^{V}\mathbf{H}, \mathbf{W}_h^{V}\mathbf{H}). \end{gathered} \tag{4}$$

Wherein, $head_h$ is the attention scores at the h-th head and $Concat()$ denotes the concatenation operation on the scores of multiple heads. In addition, $\mathbf{W}_h^{Q}$, $\mathbf{W}_h^{K}$, and $\mathbf{W}_h^{V}$ are trainable weight matrices of h-th head, which are adapted to separately map the tokens into the query, key, and value spaces.

Feed-Forward Network: FFN is a fully connected feed-forward network that operates on each position (*i.e.,* token) of sequences, which poses a vital role in existing Transformer models. Following the prior studies, we formulate the network as,

$$FFN(\hat{\mathbf{H}}) = \sigma(\hat{\mathbf{H}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{5}$$

where $\hat{\mathbf{H}}$ is the token representation learned from the self-attention block. $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are the trainable parameters of the feed-forward network, and σ is the activated function.

Residual Connection: Although the representativeness of Transformer model improves with stacking the blocks, the deeper model inevitably suffers from the problem of vanishing gradients, heavily harming the optimization of the Transformer model. Therefore, the residual connection equipped with Layer Normalization is introduced and inserted between blocks, formally,

$$\tilde{\mathbf{H}} = LayerNorm(FFN(\hat{\mathbf{H}}) + \hat{\mathbf{H}}), \tag{6}$$

where $LayerNorm()$ denotes the layer normalization operation and the output $\tilde{\mathbf{H}}$ will be the input of the subsequent block.

3 METHODOLOGY

3.1 LightGT Model

In the LightGT model, we extend the vanilla Transformer model by equipping with: (1) the modal-specific embedding that encodes the inputs of LightGT model in different modalities, (2) layer-wise position encoder which enriches the tokens with their roles in the historical interactions, and (3) light self-attention block that can be stacked to model the correlation between the tokens for user preference modeling. By performing the LightGT model in different modalities, we can learn the modal-specific user preference on the content information for user-item interaction prediction.

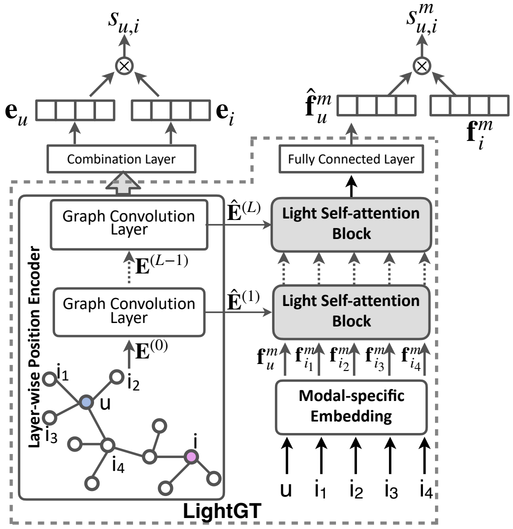


Figure 1: An illustration of our proposed model.

3.1.1 *Modal-specific Embedding.*

In the Transformer-based model, it is essential to construct the input tokens for self-attention scoring. Here, our LightGT model targets at discovering the user preference from her/his historical interactions. We hence take the user and her/his interacted items as tokens of the model. It not only builds the correlation between the co-interacted items but distills the informative signal under the supervision of user-item interactions.

Inspired by the prior study [35], we embed the tokens in each modality, in order to learn the modal-specific user preference. In particular, taking user u in Figure 1 and her/his interacted items (*i.e.,* i_1, i_2, i_3, and i_4) as an example, we can represent them as:

$$\mathbf{F}_u^{m} = [\mathbf{f}_u^{m}, \mathbf{f}_{i_1}^{m}, \mathbf{f}_{i_2}^{m}, \mathbf{f}_{i_3}^{m}, \mathbf{f}_{i_4}^{m}], \tag{7}$$

where $\mathbf{f}_{i_1}^{m} \in \mathbb{R}^{d\times 1}$, $\mathbf{f}_{i_2}^{m} \in \mathbb{R}^{d\times 1}$, $\mathbf{f}_{i_3}^{m} \in \mathbb{R}^{d\times 1}$, and $\mathbf{f}_{i_4}^{m} \in \mathbb{R}^{d\times 1}$ are the features of item i_1, i_2, i_3, and i_4 in modality m, respectively. They are captured from the content information by a pre-trained feature extractor and compressed into d-dimensional vectors. Moreover, $\mathbf{f}_u^{m} \in \mathbb{R}^{d\times 1}$ denotes the trainable representation of user u in modality m. After gathering these vectors together, we yield the modal-specific tokens' representations $\mathbf{F}_u^{m}$ for user preference modeling.

For the sake of notational brevity, in what follows, we conduct the same operations on different modalities and omit indicator m without specific clarification.

3.1.2 *Layer-wise Position Encoder.*

Typically, position information plays a vital role in the Transformer model, which provides the prior knowledge to enhance the attention measurement. Unlike the sequential inputs, *e.g.,* words in sentences and frames in videos, our obtained tokens are not aware of the position in the sequence. As such, we resort to use the structure information of the user-item graph to encode the positions of tokens, which reflects their relations in the historical user-item interactions.

Towards this end, we define the id embeddings of users and items, and then reorganize the user-item interactions in the history as a bipartite graph:

$$\mathcal{G} = \{\mathbf{E}, \mathbf{A}\}. \tag{8}$$

Whereinto, $\mathbf{E} \in \mathbb{R}^{(N+M)\times d}$ is the trainable embedding matrix for the user and item nodes. And, $\mathbf{A} \in [0,1]^{(N+M)\times(N+M)}$ is the adjacency matrix, functioning as the edges of the bipartite graph. More specifically, an edge $\mathbf{A}_{ui} = 1$ indicates an observed interaction between user u and item i; otherwise $\mathbf{A}_{ui} = 0$.

According to the construction of the interaction graph, the structure information can reflect the affinities between the nodes in the recommendation scenario. As shown in Figure 1, given a pair of item nodes locating in the two-hop neighbors (*e.g.,* i_2 and i_3), we can find they are interacted by at least one user node. Compared with the node pairs being distant in the graph (*e.g.,* i_2 and i), the nodes with more common neighbors tend to have more similar representations. Therefore, we can leverage the representations of structure information learned from graph to encode the position of corresponding nodes (*i.e.,* item and user).

To encode the positions, we employ the GCN model on the bipartite graph for the structure information modeling. In particular, by iteratively performing the stacked graph convolutional layers, we can pass the message from multi-hop neighbors into centric nodes and thus represent their structure information with the neighbors' embeddings. Taking the l-th layer as an example, we model the nodes' structure information by aggregating their neighbors' embeddings, formally,

$$\begin{aligned} \mathbf{e}_u^{(l)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_i^{(l-1)}, \\ \mathbf{e}_i^{(l)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(l-1)}, \end{aligned} \tag{9}$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the sets of neighbor nodes connecting to user u and item i, respectively. $|\mathcal{N}_u|$ and $|\mathcal{N}_i|$ are the number of nodes in two sets. The symmetric normalization term $\frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}}$ is able to avoid the scale of embeddings when stacking the graph convolutional layers. $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ are the structure information learned from the l-hop neighbors of user u and item i. Note that we set $\mathbf{e}_u^{(0)}$ as the embedding of user u, *i.e.,* $\mathbf{e}_u \in \mathbf{E}$; analogically, $\mathbf{e}_i^{(0)}$ equals to $\mathbf{e}_i \in \mathbf{E}$.

Thereafter, we implement the layer-wise position encoder that absorbs the structure information and generates the position encodings for the different self-attention blocks, formally,

$$\begin{cases} \hat{\mathbf{E}}^{(1)} = \frac{1}{2}(\mathbf{E}^{(0)} + \mathbf{E}^{(1)})), \\ \hat{\mathbf{E}}^{(2)} = \frac{1}{3}(\mathbf{E}^{(0)} + \mathbf{E}^{(1)} + \mathbf{E}^{(2)})), \\ \quad \cdots, \\ \hat{\mathbf{E}}^{(L)} = \frac{1}{L+1}(\mathbf{E}^{(0)} + \mathbf{E}^{(1)} + \cdots + \mathbf{E}^{(L)}), \end{cases} \tag{10}$$

where $\mathbf{E}^{(l)}$ denotes the representations of structure information learned from l-th graph convolutional layer. $\hat{\mathbf{E}}^{(l)}$ represents the position encoding for the l-th self-attention block. The motivation of this position encoder is that we need more sufficient structure information to help the self-attention scoring when we stack the self-attention blocks.

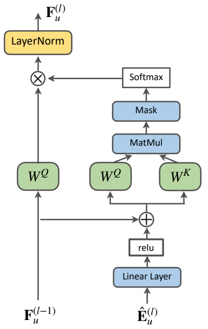


Figure 2: An illustration of the *l*-th light self-attention block.

Furthermore, we consider the low confidence of tokens' representations at the beginning of the training phase, due to their irrelevant signal learned from the off-the-shelf extractors. To alleviate their negative feedback to the structural information modeling, we further introduce a stop-gradient operation into the position encoder, formally,

$$\hat{\mathbf{E}}^{(l)} = stopgrad(\hat{\mathbf{E}}^{(l)}) \tag{11}$$

where $stopgrad()$ represents the stop-gradient operation to cut off the back propagation from the self-attention blocks to the user and item embeddings.

Overall, from the perspective of recommendation algorithm, the layer-wise position encoder introduces the CF-based similarities between nodes into the self-attention blocks at different levels. With such a position encoder, we can optimize the self-attention measurement by incorporating the external knowledge of user-item interactions with tokens' features, in order to discover the features relevant to the user preferences.

3.1.3 Light Self-attention Block. Recalling the prior studies, the promising performance of Transformer has been witnessed in various applications ranging from the natural language processing to computer vision. However, we find that it is hard to optimize the standard Transformer efficiently and accurately, due to the data overload and sparsity interactions in the recommender system. After experimentally analyzing the components in the vanilla Transformer, we devise a light self-attention block by simplifying the heavy and burdensome components.

To be implemented, we first embed the tokens' position encodings $\hat{\mathbf{E}}_u^{(l)}$ into their representation. As Figure 2 displays, in the l-th block, we formulate the position-enriched representations of tokens as:

$$\mathbf{H}_u^{(l)} = sigmoid(\mathbf{W}_m \hat{\mathbf{E}}_u^{(l)} + \mathbf{b}_m) \oplus \mathbf{F}_u^{(l-1)}, \tag{12}$$

where $\oplus$ represents the element-wise addition operation. $\mathbf{F}_u^{(l-1)}$ denotes the tokens' representations learned from the previous block. As for the first block, $\mathbf{F}_u^{(0)}$ is the off-the-shelf features of tokens, *i.e.,* $\mathbf{F}_u^m$. To fuse these two vectors, we project the position encoding to m modality with a linear layer equipped with the activation function $sigmoid()$, where $\mathbf{W}_m \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_m \in \mathbb{R}^{d\times 1}$ are the trainable parameters in modality m. And then, we can achieve the position-enriched token representations at l-th layer, *i.e.,* $\mathbf{H}_u^{(l)}$.

With the enriched representations of tokens, we can obtain the query and key vectors for similarity measurement, formally,

$$\mathbf{Q} = \mathbf{W}^{Q}\mathbf{H}_u^{(l)}, \mathbf{K} = \mathbf{W}^{K}\mathbf{H}_u^{(l)}, \tag{13}$$

where $\mathbf{W}^K \in \mathbb{R}^{d \times d}$, and $\mathbf{W}^Q \in \mathbb{R}^{d \times d}$ are the trainable parameters to map the tokens' representations into the query and key spaces, respectively.

Next, we empirically remove the FFN from each self-attention block and residual connection between the blocks. More importantly, we find that the single-head attention module can achieve a comparative performance with the multi-head attention one. Therefore, we conduct the scaled self-attention operations:

$$\mathbf{F}_u^{*} = softmax(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}})\mathbf{W}^{V}\mathbf{F}_u^{(l-1)}, \tag{14}$$

where $\mathbf{F}_u^*$ is the tokens' representations enhanced by aggregating the context information. $\mathbf{W}^V \in \mathbb{R}^{d \times d}$ represents the parameter to gain the value vectors. It is worth noting that we instead use the tokens' representations without the position encodings (*i.e.,* $\mathbf{F}_u^{(l-1)}$). This is because that we focus on the user preference on the content information and avoid the effects from the id embeddings.

Thereafter, we employ the normalization operation and generate the tokens' representation at l-th self-attention block:

$$\mathbf{F}_u^{(l)} = LayerNorm(\mathbf{F}_u^{*}), \tag{15}$$

where $\mathbf{F}_u^{(l)}$ is the output matrix of tokens at l-th block and $LayerNorm()$ denotes the layer normalization function.

Taking advantages of the self-attention mechanism, we model the relation of user-item and item-item pairs. Thereinto, the user-item affinities can help to emphasize the items' features users prefer, and the item-item correlation highlight the co-occurrence information affecting the user's decision-making.

3.1.4 *User preference Modeling.*

Stacking L self-attention blocks, we obtain the enhanced representations of input items and user tokens. For the users, some informative cues are captured at different blocks and aggregated with their representations. To model the user's preference on the item content, we formulate a fully connected network (FCN) to re-express user u:

$$\hat{\mathbf{f}}_u = leaky\_relu(\mathbf{W}_m^{'}\mathbf{f}_u^{(L)} + \mathbf{b}_m^{'}), \tag{16}$$

where $\mathbf{f}_u^{(L)} \in \mathbf{F}_u^{(L)}$ is u's representation after the L-th light self-attention block. $\mathbf{W}_m^{'}$, $\mathbf{b}_m^{'}$, and $leaky\_relu$ are the projection matrix, bias vector, and Leaky_relu function [24], respectively. Ultimately, we obtain the enhanced representation of user preference $\hat{\mathbf{f}}_u$.

3.2 Model Prediction

In order to predict the interactions between users and items, we follow the common solution of multimedia recommendation [3, 12, 35] and separately estimate their CF-based and content-based similarities.

In particular, we are able to achieve the CF-based embeddings of users and items based on the graph convolutional networks. By combining the structure information (*a.k.a.* collaborative signal) learned from L layers, we can inject the high-order interaction information into the embeddings [13]. We formulate the combination layer as:

$$\mathbf{e}_u = \sum_{l=0}^{L} \mathbf{e}_u^{(l)}, \mathbf{e}_i = \sum_{l=0}^{L} \mathbf{e}_i^{(l)}. \tag{17}$$

Then, we can measure the similarities between users and items upon their CF-based representations, as,

$$s_{u,i} = \mathbf{e}_u \mathbf{e}_i^{\top}, \tag{18}$$

where $s_{u,i}$ is our desired score estimating how likely user u prefers item i according to the historical behaviours.

As for the content-based score, we gain the item features from the off-the-shelf features and calculate the similarity between the user and target item on modality m, formally,

$$s_{u,i}^{m} = \hat{\mathbf{f}}_u^{m}(\mathbf{f}_i^{m})^{\top}. \tag{19}$$

Whereinto, $s_{u,i}^m$ scores how user u prefers the content of item i.

With the obtained CF- and content-based scores, we can combine them to predict the user-item interaction, formally,

$$s_{u,i}^{*} = \lambda s_{u,i} + (1-\lambda)\sum_{m \in \mathcal{M}} s_{u,i}^{m}, \tag{20}$$

where λ is a hyper-parameter to balance the two types of information in the final interaction score $s_{u,i}^*$.

3.3 Optimization

To optimize the parameters of our proposed model, we apply Bayesian Personalized Ranking (BPR) [26], which is a well-known pairwise ranking optimization algorithm, and reconstruct the historical interactions between users and items.

Towards this end, we organize the triplet of one user, one item she/he observed, and one item unobserved by the user. It can be defined as,

$$\mathcal{R} = \{(u,i,i') | (u,i) \in O, (u,i') \notin O\}, \tag{21}$$

where $\mathcal{R}$ is a set of triples for training. According to the assumption that the user prefers the observed item rather than the unobserved one, the objective function of BPR can be formulated as,

$$\mathcal{L} = \sum_{(u,i,i') \in \mathcal{R}} -\ln \varphi(s_{u,i}^{*} - s_{u,i'}^{*}) + \eta \left\|\Theta\right\|_2^2, \tag{22}$$

where $\varphi(\cdot)$, η, and Θ denote the *sigmod* function, regularization weight, and trainable parameters of our proposed model, respectively.

4 EXPERIMENT

To evaluate the effectiveness of our proposed model, we conduct extensive experiments on three public datasets and answer the following research questions:

- **RQ1** How does our proposed model perform compared with state-of-the-art bundle recommendation models?
- **RQ2** How do the designs (*i.e.* layer-wise position encoder and light self-attention block) in LightGT affect the performance of our model?
- **RQ3** What is the effectiveness and efficiency of the components (*i.e.,* multi-head attention, FFN, and residual connection) in the self-attention block?

Table 1: Statistics of the evaluation datasets. (d_v, d_a, and d_t denote the dimensions of visual, acoustic, and textual modality feature data, respectively.)

Dataset	#Users	#Items	#Inter	d_v	d_a	d_t
MovieLens	55,485	5,986	1,239,508	2048	128	100
TikTok	36,656	76,085	726,065	128	128	128
Kwai	7,010	86,483	298,492	2048	-	128

- **RQ4** Can LightGT learn the superior representation of user preference?

Before answering the above four questions, we describe the datasets, evaluation protocols, baselines, and parameter settings in the experiments.

4.1 Experiment Settings

4.1.1 Dataset. To test the effectiveness of our proposed model, we follow the prior studies and conduct extensive experiments on three datasets, including Movielens, Tiktok, and Kwai, which are widely used in current research on multimedia recommendation. More detailed information on these datasets could be found in Table **??**.

- **Movielens** Movielens dataset[1] are designed for the studies of recommender systems. In terms of multimedia recommendation, the authors of MMGCN collected the videos' trailers, titles, and textual descriptions. With the pre-trained models (*e.g.* ResNet [11], VGGish [14], and Sentence2Vector [1]), the visual, acoustic, and textual features are extracted from the frames, audio tracks, and descriptions, respectively.
- **Tiktok** The items in this dataset contain multimedia features, involving visual, acoustic, and textual modalities. According to the publisher of Tiktok[2], the features are extracted by some deep learning models pre-trained on non-recommendation datasets.
- **Kwai** Similar to the above two datasets, the items in the Kwai dataset consist of multi-modal features obtained by the feature extractors. The extractors are also designed for the recognition tasks on different modalities and trained with datasets irrelevant to recommendation tasks.

4.1.2 Evaluation Protocols. For each dataset, we randomly split the interaction records per user into the training, validation, and testing sets under the ratio $8:1:1$. The validation set and testing set are respectively used to tune the hyper-parameters and evaluate the performance in the experiments. Moreover, following the widely-used evaluation metrics [13, 32], we adopted recall@K (R@10) and Normalized Discounted Cumulative Gain (N@K) to evaluate the performance of methods. By default, we set $K = 10$ and reported the average values of the two metrics for all users in the test set. We make use of Nvidia Tesla V100 graphics card (32GB Memory) for all of the experiments.

4.1.3 Baselines. We compare our proposed model with the state-of-the-art baselines, including the CF-based recommendation models (*i.e.*, *GraphSAGE*, *NGCF*, *GAT*, and *LightGCN*) and multimedia recommendation models (*i.e.*, *VBPR*, *MMGCN*, *GRCN*, and *LATTICE*)[3]. In particular,

- **GraphSAGE** [10]: This model conducts the graph convolutional operations on the user-item graph and injects the structure information of nodes into their representations, in order to optimize the predictions of user-item interactions.
- **NGCF** [32]: This model presents a recommendation framework to explicitly integrate the CF signal into the user and item embeddings. By exploiting the high-order connectivity from user-item interactions, the model encodes more effective CF signals into the representations.
- **GAT** [29]: The method aims to learn the weights for user-item interaction, *i.e.,* edges in the user-item graph, according to the similarity of node pairs. It adaptively aggregate the neighbors' messages of each node and enhance the node's representation for interaction predictions.
- **LightGCN** [13]: LightGCN is a popular GCN-based recommendation model. By removing the feature transformation and nonlinear activation, a light graph convolution layer is designed to effectively and efficiently model the high-order interaction information into the user and item embeddings.
- **VBPR** [12]: This is a benchmark model in the multimedia recommendation. It incorporates multi-modal features extracted by pre-trained models with matrix factorization framework to predict the interactions between users and items.
- **MMGCN** [35]: MMGCN is the first GCN-based multimedia recommendation model, which constructs the user-item graph in each modality and learns the modal-specific user preference with graph convolutional operations. It directly takes the extracted features as items' representation and accordingly learns the user preference from the historical items.
- **GRCN** [34]: This model focuses on refining the structure of the user-item graph to alleviate the bias caused by false positive interaction records. Hence, the model leverages the extracted features to measure the similarities between users and items, and then effectively learns user preference based on the off-the-shelf features.
- **LATTICE** [40]: This method is designed to model the correlation between the items by computing the cosine distances of their features, in order to enrich the CF-based signal with the semantic similarities for the recommendation.

4.1.4 Parameter Settings. We use the Pytorch[4] and torch-geometric packages[5] to implement our proposed model. In particular, we initialize the parameters of model with Xavier algorithm [9] and optimize them with Adam optimizer [17]. For the learning rate and regularization weight, we conduct the grid search in $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and $\{0.00001, 0.0001, 0.001, 0.01, 0.1\}$, respectively. When recall@10 on the validation data does not increase for 20 successive epochs, we stopped the training and reported the results on testing dataset. For the baselines, we followed the designs in their articles to achieve the best

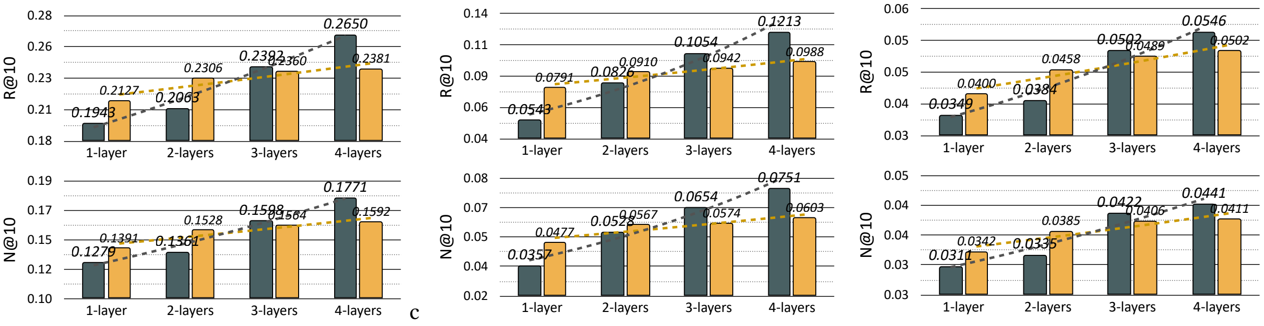
[1] https://movielens.org/.

[2] https://www.tiktok.com/.

[3] Here, we omit ACF model, since the raw data, like keyframes and soundtrack, is not provided in the Kwai and Tiktok datasets.

[4] https://pytorch.org/.

[5] https://pytorch-geometric.readthedocs.io/.



(a) Recall@10 and NDCG@10 on Movielens (b) Recall@10 and NDCG@10 on Tiktok (c) Recall@10 and NDCG@10 on Kwai

Figure 3: Results of LightGT and LightGCN in terms of different layers on Movielens, Tiktok, and Kwai datasets. (Green and yellow colors remark LightGT and LightGCN, respectively. Best viewed in color.)

Table 2: Overall performance comparison between our model and the baselines on three datasets.

Methods	MovieLens		Tiktok		Kwai	
	R@10	N@10	R@10	N@10	R@10	N@10
GraphSAGE	0.2129	0.1388	0.0778	0.0476	0.0424	0.0344
NGCF	0.2340	0.1383	0.0906	0.0547	0.0427	0.0363
GAT	0.2342	0.1589	0.0945	0.0575	0.0441	0.0369
LightGCN	0.2381	0.1592	0.0988	0.0603	0.0502	0.0411
VBPR	0.1927	0.1207	0.0600	0.0397	0.0302	0.0221
MMGCN	0.2453	0.1523	0.1645	0.0579	0.0456	0.0374
GRCN	0.2520	0.1683	0.0952	0.0584	0.0473	0.0403
LATTICE	0.2520	0.1680	0.0984	0.0611	0.0483	0.0400
LightGT	**0.2650**	**0.1771**	**0.1213**	**0.0751**	**0.0546**	**0.0441**
% Improv.	5.16%	5.23%	22.77%	22.91%	8.77%	7.30%
p-value	1.34e-4	3.01e-4	6.49e-7	8.51e-6	8.71e-5	1.84e-4

Table 3: Overall performance comparison between our model and the baselines on three datasets.

Methods	MovieLens		Tiktok		Kwai	
	R@10	N@10	R@10	N@10	R@10	N@10
w/o PE	0.2604	0.1753	0.1192	0.0729	0.0513	0.0430
w/o SG	0.2599	0.1738	0.1074	0.0644	0.0531	0.0435
w/o LW	0.2627	0.1767	0.1207	0.0749	0.0530	0.0435
LightGT	0.2650	0.1771	0.1213	0.0751	0.0546	0.0441

performance. For fairness, we did the same options and fixed the dimension of the ID embedding vector to 64.

4.2 Performance Comparison (RQ1)

To demonstrate the effectiveness of our proposed model, we conduct the experiments on three datasets and list the results of LightGT and baselines in Table 2. According to the results, we have the following observations:

- On three datasets, our proposed LightGT substantially outperforms all the baselines by a large margin. In particular, LightGT improves the strongest baselines *w.r.t.* Recall by 5.16%, 22.77%, and 8.77%, on the three datasets respectively. Such improvements verify the effectiveness of our proposed model. We attribute the performance to the items’ informative features extraction based on their correlation modeling, so as to optimize the representations of user preference on the content information. Moreover, p-value < 0.05 shows that the improvements in our model are statistically significant.
- LightGT and the strongest baseline LATTICE perform better than the other baselines in most cases. It indicates that explicitly modeling the correlation between items can help the prediction of user-item interaction. Further, LATTICE clearly underperforms our LightGT in terms of two metrics. It makes sense since LATTICE learns the user preference based on the raw features learned by the pre-trained extractor, while our LightGT captures the informative cues from the raw ones for user preference modeling.
- Compared with content-based models, CF-based recommendation models achieve poor performance in most cases. The reason for their suboptimal results might be the lack of the content-based similarity of the user-item pair. This verifies the reasonability of the motivation and designs of our proposed model again.
- Cross three datasets, it is easy to find that the improvements of LightGT on Tiktok and Kwai datasets are larger than that on Movielens. We suggest that this might be caused by their different densities, *i.e.,* interactions per user. Hence, the performance on the sparser datasets can be improved larger by the superior content-based similarity.

4.3 Ablation Study (RQ2)

4.3.1 Effect of Layer-wise Position Encoder. To test the layer-wise position encoder, we compare LightGT with several variant models:

w/o PE: We remove the position encoder and conduct the light self-attention blocks merely on the content features.

w/o SG: In this variant, we remove the stop-gradient operation from the layer-wise position encoder, *i.e.,* Eq. 11.

w/o LW: Removing the layer-wise designs, we calculate the position encoding in each block with the average of representations of structure information learned by all LGC layers.

Table 4: Performance comparison between single- and multi-head self-attention blocks on three datasets. (Time (s) denotes the seconds per epoch. Head # represents the number of heads in the self-attention block.)

	Head #	R@10	N@10	Time (s)
Movielens	1 Head	0.2650	0.1771	167s
	2 Heads	0.2621 (-1.09%)	0.1753 (-1.02%)	229s
	4 Heads	0.2613 (-1.40%)	0.1741 (-1.70%)	334s
Tiktok	1 Head	0.1213	0.0751	80s
	2 Heads	0.1113 (-8.24%)	0.0695 (-7.46%)	106s
	4 Heads	0.1095 (-9.72%)	0.0683 (-2.24%)	164s
Kwai	1 Head	0.0546	0.0441	35s
	2 Heads	0.0515 (-5.68%)	0.0423 (-4.08%)	47s
	4 Heads	0.0523 (-4.21%)	0.0427 (-3.17%)	61s

According to the results listed in Table 3, we find that our proposed LightGT consistently achieves superior performance compared with three variant models. In particular, after removing the position encoder from the model (*i.e.,* **w/o PE**), the performance drops on three datasets, which verifies the effectiveness of position encoding in the Transformer-based model. Moreover, conducting the layer-wise position encoder without the stop-grained operations(**w/o SG**) degrades the performance clearly. The reason might be that the suboptimal item features bias the CF embeddings by passing the noisy signal during the back-propagation, especially at the beginning of the training phase. As for the variant model **w/o LW**, its inferior performance indicates that the encodings may be inconsistent in different blocks and empirically verifies the layer-wise design in the position encoder.

4.3.2 Effect of Light Self-attention Block. For the light self-attention block, we test the layer (*i.e.,* block) numbers in the range of {1, 2, 3, 4}. As a comparison, we conduct the LightGCN under the same setting.

As illustrated in Figure 3, we observe that our LightGT significantly outperforms LightGCN on three different datasets when the number of layers is larger than two. Furthermore, by increasing the layer number, the improvements achieved by LightGT are more significant than that of LightGCN. From the observations, we conclude the findings that: 1) the deeper GCNs in our model can model and inject more comprehensive structure information into the tokens' representations, further enhancing the self-attention scoring; and 2) stacked light self-attention blocks are capable of distilling the effective features from the off-the-shelf ones. This again verifies the effectiveness and reasonability of the designs in our proposed model.

4.4 In-depth Analysis (RQ3)

4.4.1 Effect of Multi-head Attention. To investigate the multi-head attention in the Transformer-based recommendation model, we compare 2-head and 4-head self-attention blocks with our LightGT on three datasets.[6]

We list their results *w.r.t.* R@10, N@10, and Times in Table 4 and have the observation that LightGT consistently gains the best performance on all the datasets. Specifically, the values of recall@10 and NDCG@10 decrease with increasing the number of heads, while the time cost raises sharply. It figures out that the multi-head self-attention is helpful in computer vision and natural language processing tasks, but will harm the performance of the recommender system. The reason might be that it is enough to model the tokens' correlation in a single aspect.

Table 5: Performance comparison between LightGT and the baseline equipped with FFN (w/ FFN) on three datasets.

	Methods	R@10	N@10	Time (s)
Movielens	LightGT	0.2650	0.1771	167s
	w/ FFN	0.2474 (-6.64%)	0.1645 (-7.11%)	255s
Tiktok	LightGT	0.1213	0.0751	80s
	w/ FFN	0.1148 (-5.36%)	0.0709 (-5.59%)	154s
Kwai	LightGT	0.0546	0.0441	35s
	w/ FFN	0.0508 (-6.96%)	0.0424 (-3.85%)	79s

Table 6: Performance comparison between LightGT and the baseline equipped with residual connection (w/ RC) on three datasets.

	Methods	R@10	N@10	Time (s)
Movielens	LightGT	0.2650	0.1771	167s
	w/ RC	0.2606 (-1.66%)	0.1750 (-1.19%)	188s
Tiktok	LightGT	0.1213	0.0751	80s
	w/ RC	0.1135 (-6.43%)	0.0705 (-6.13%)	112s
Kwai	LightGT	0.0546	0.0441	35s
	w/ RC	0.0496 (-9.15%)	0.0405 (-8.16%)	67s

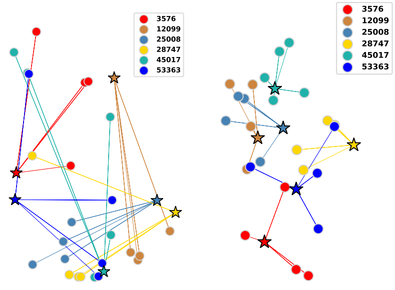
4.4.2 Effect of FFN. Considering the effectiveness of FFN in vanilla Transformer, we implement it in our light self-attention block to test its necessity in recommendation tasks. For this goal, we conduct the experiments on three datasets and record its results *w.r.t.* R@10 and N@10 as well as the time consumption in Table 5.

Observing the results, we surprisingly find that the performance largely drops when we implement FFNs in the self-attention block (*i.e.,* **w/ FFN**). We attribute it to the overfitting problem, since the FFNs introduce some parameters to be learned. More specifically, unlike the tasks in the computer vision and natural language processing domains, sufficiently modeling the semantic information might be unnecessary in the recommendation application. Jointly analyzing the time consumption of the two models, we easily find that the light self-attention block benefits both the effectiveness and efficiency of our Transformer-based model.

4.4.3 Effect of Residual Connection. To validate the residual connection, we conduct the experiments on three datasets by equipping the residual connection between two successive self-attention blocks, *i.e.,* **w/ RC**. Beyond the recall, and NDCG metrics, we report the time cost of each epoch.

From the results shown in Table 6, we observe that LightGT achieves comparative performance *w.r.t.* R@10 and N@10 with the baseline model. Even without the residual connection, unexpectedly, our LightGT outperforms the baseline on three datasets by a margin. More importantly, compared with the baseline, we find that LightGT costs significantly less time per epoch, which comes from removing

[6] We only test 2- and 4-head self-attention variants due to memory limitations of the GPUs we used.



(a) MMGCN (b) LightGT

Figure 4: Visualization of the t-SNE representations of user and items learned from MMGCN and LightGT. (Stars denote the users and the points with the same color are the relevant items. Best viewed in color.)

residual connections. Overall, the observation verifies that our design not only has a slight effect on the accuracy but improves the efficiency of model training and inference.

4.5 Visualization (RQ4)

In this section, we explore whether the representations derived from our proposed model are better to measure the user preference for the items' content information. Towards this end, we randomly selected six users from Tiktok dataset. After collecting several items that they are never paired in the training phase, we conduct the t-SNE algorithm to plot their representations on a two-dimension space and visualize the item and user representations learned from MMGCN and LightGT models, as shown in Figure 4.

Compared with the MMGCN model, we observe that the stars and points with the same color, *i.e.,* the user and her/his relevant items, are embedded nearer when performing the LightGT model. These discernible clusters derived from the LightGT model show that our model effectively model the user preference. We attribute it to that LightGT model is capable of capturing the informative signal from the items' features. It verifies the effectiveness of our proposed model again.

5 RELATED WORK

5.1 Multimedia Recommender System

In terms of multimedia recommender systems, pioneer researchers are prone to incorporating CF-based methods with content-based recommendation models [2, 20, 30, 33]. Through restoring user-item interactions in the history, they target at modeling the user preference on item contents to generate a high-quality recommender system. For instance, He *et al.,* proposed Visual Bayesian Personalized Ranking (VBPR) approach [12], which extends BPR method and models the user preference on the visual information. After that, Chen *et al.,* developed an attention-based recommendation model, termed Attentive Collaborative Filtering (ACF) [3], to learn the user interests on both item- and component-level in the multimedia recommendation. Similarly, Liu *et al.* [22] presented a User-Video Co-Attention Network (UVCAN) for micro-video recommendation tasks, in order to capture multi-modal information from both the user and micro-video side. Considering the users' different tastes in the different modalities, Wei *et al.* [35] introduced GCNs into the multimedia recommendation model and proposed a Multi-modal Graph Convolution Network. Wherein, it is able to capture the model-specific user preference and optimize the item representations simultaneously.

Although these methods achieve state-of-the-art results, they overestimate the importance of feature extraction in the multimedia recommendation. In this work, we consider the correlation between the items within the user's historical interaction and distill the item features captured from the pre-trained extractors, so as to effectively model the user preference for content information.

5.2 Transformer-based Recommendation Model

Recent years have witnessed the prevalence and effectiveness of the Transformer model in various domains ranging from natural language processing [7, 8, 18] to computer vision [5, 15, 16]. Taking advantage of the contextualized information modeling, the Transformer model recently draw much research attention from the recommendation domain to explore the relation among items. For instance, the prior studies [4, 6, 27, 36–38] construct the user-item interactions in history as the sequential structure and feed them into the Transformer-based recommendation methods to model user behavior sequences. Different from these sequential recommendation models, Liu *et al.,* [23] proposed a novel pre-trained multimodal graph transformer model to learn item representations by considering both item side information and their relationships. More recently, Li *et al.,* [19] developed a novel transformer-based model, named PEAR, for re-ranking. It not only captures interactions at feature- and item- levels, but models item contexts from the historically clicked item list.

However, these methods mostly inherit the designs of the vanilla Transformer model and leave their effectiveness and efficiency untouched. In our proposed model, we explore these operations and devise a light Transformer architecture for the multimedia recommendation.

6 CONCLUSION

In our work, we propose a new Transformer-based model, named Light Graph Transformer (LightGT), for multimedia recommendation. Within the proposed model, we design a modal-specific embedding and a layer-wise position encoder for effective features distillation, and devise a light self-attention block for efficient self-attention scoring. By modeling the correlation between items obaserved by the same user, we achieve the superior user preference on item content for user-item interaction prediction. Observing the experimental results on three datasets, we demonstrate the effectiveness and efficiency of our LightGT.

7 ACKNOWLEDGEMENT

This research is supported by the National Natural Science Foundation of China No.: 9227010114, No.:U1936203 and No.:62236003; Shenzhen College Stability Support Plan, No.: GXWD20220817144428005; and the University Synergy Innovation Program of Anhui Province (GXXT-2022-040).

REFERENCES

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*. 1–16.

[2] Feiyu Chen, Junjie Wang, Yinwei Wei, Hai-Tao Zheng, and Jie Shao. 2022. Breaking Isolation: Multimodal Graph Fusion for Multimedia Recommendation by Edge-wise Modulation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 385–394.

[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 International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.

[4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[6] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 433–442.

[7] Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2022. LasUIE: Unifying Information Extraction with Latent Adaptive Structure-aware Generative Language Model. In *Proceedings of the Advances in Neural Information Processing Systems, NeurIPS 2022*. 15460–15475.

[8] Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7014–7026.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and statistics*. 249–256.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[11] 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.

[12] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 144–150.

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.

[14] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 131–135.

[15] Wei Ji, Long Chen, Yinwei Wei, Yiming Wu, and Tat-Seng Chua. 2022. MRTNet: Multi-Resolution Temporal Network for Video Sentence Grounding. *arXiv preprint arXiv:2212.13163* (2022).

[16] Wei Ji, Renjie Liang, Zhedong Zheng, Wenqiao Zhang, Shengyu Zhang, Juncheng Li, Mengze Li, and Tat-seng Chua. 2023. Are Binary Annotations Sufficient? Video Moment Retrieval via Hierarchical Uncertainty-based Active Learning. (2023).

[17] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*. 1–16.

[18] Svea Klaus, Ria Van Hecke, Kaweh Djafari Naini, Ismail Sengor Altingovde, Juan Bernabé-Moreno, and Enrique Herrera-Viedma. 2022. Summarizing Legal Regulatory Documents using Transformers. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2426–2430.

[19] Yi Li, Jieming Zhu, Weiwen Liu, Liangcai Su, Guohao Cai, Qi Zhang, Ruiming Tang, Xi Xiao, and Xiuqiang He. 2022. PEAR: Personalized Re-ranking with Contextualized Transformer for Recommendation. In *Proceedings of the International Conference on World Wide Web*.

[20] Fan Liu, Huilin Chen, Zhiyong Cheng, Anan Liu, Liqiang Nie, and Mohan Kankanhalli. 2022. Disentangled Multimodal Representation Learning for Recommendation. *IEEE Transactions on Multimedia* (2022), 1–11.

[21] 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.

[22] 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.

[23] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. 2021. Pre-training graph transformer with multimodal side information for recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2853–2861.

[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the international conference on machine learning*. 3–9.

[25] Yanjun Qin, Yuchen Fang, Haiyong Luo, Fang Zhao, and Chenxing Wang. 2022. Next Point-of-Interest Recommendation with Auto-Correlation Enhanced Multi-Modal Transformer Network. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2612–2616.

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

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. 1–12.

[30] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

[31] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*. 3562–3571.

[32] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 165–174.

[33] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5382–5390.

[34] 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.

[35] 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.

[36] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*. 328–337.

[37] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2397–2406.

[38] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4486–4493.

[39] Enming Yuan, Wei Guo, Zhicheng He, Huifeng Guo, Chengkai Liu, and Ruiming Tang. 2022. Multi-Behavior Sequential Transformer Recommender. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1642–1652.

[40] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining Latent Structures for Multimedia Recommendation. In *Proceedings of ACM International Conference on Multimedia*. 3872–3880.

[41] Jie Zou, Evangelos Kanoulas, Pengjie Ren, Zhaochun Ren, Aixin Sun, and Cheng Long. 2022. Improving conversational recommender systems via transformer-based sequential modelling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2319–2324.