Graph Neural Network for Tag Ranking in Tag-enhanced Video Recommendation

Qi Liu*
WeChat, Tencent
addisliu@tencent.com

Ruobing Xie*
WeChat, Tencent
ruobingxie@tencent.com

Lei Chen
WeChat, Tencent
collinschen@tencent.com

Shukai Liu
WeChat, Tencent
shukailiu@tencent.com

Ke Tu
Tsinghua University
tuke1993@gmail.com

Peng Cui
Tsinghua University
cuip@tsinghua.edu.cn

Bo Zhang
WeChat, Tencent
nevinzhang@tencent.com

Leyu Lin
WeChat, Tencent
goshawklin@tencent.com

ABSTRACT

In tag-enhanced video recommendation systems, videos are attached with some tags that highlight the contents of videos from different aspects. Tag ranking in such recommendation systems provides personalized tag lists for videos from their tag candidates. A better tag ranking model could attract users to click more tags, enter their corresponding tag channels, and watch more tag-specific videos, which improves both tag click rate and video watching time. However, most conventional tag ranking models merely concentrate on tag-video relevance or tag-related behaviors, ignoring the rich information in video-related behaviors. We should consider user preferences on both tags and videos. In this paper, we propose a novel Graph neural network based tag ranking (GraphTR) framework on a huge heterogeneous network with video, tag, user and media. We design a novel graph neural network that combines multi-field transformer, GraphSAGE and neural FM layers in node aggregation. We also propose a neighbor-similarity based loss to encode various user preferences into heterogeneous node representations. In experiments, we conduct both offline and online evaluations on a real-world video recommendation system in WeChat Top Stories. The significant improvements in both video and tag related metrics confirm the effectiveness and robustness in real-world tag-enhanced video recommendation. Currently, GraphTR has been deployed on WeChat Top Stories for more than six months. The source codes are in https://github.com/lqfarmer/GraphTR.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

tag ranking; graph neural network; heterogeneous network

ACM Reference Format:
Qi Liu, Ruobing Xie, Lei Chen, Shukai Liu, Ke Tu, Peng Cui, Bo Zhang, and Leyu Lin. 2020. Graph Neural Network for Tag Ranking in Tag-enhanced Video Recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3416021

1 INTRODUCTION

Personalized recommendation system aims to provide customized items for users according to their preferences, which has been widely used in various fields [24]. Recently, video recommendation becomes more and more essential for billions of users to get information or entertainment. Differing from texts and images, videos usually contain much more information that is not explicit to users at first sight. Although titles and cover images can partially indicate the main ideas of videos, it is still hard for those static abstracts to capture different aspects of user preferences in videos.

To address this issue, many video recommendation systems such as Youtube and Netflix use tags attached to videos to highlight different user concentrations of these videos. Fig. 1 shows a classical **tag-enhanced video recommendation system** in *WeChat Top Stories*. Each video contains some tag candidates pre-labeled by human annotators, and tag ranking models provide personalized tags from these candidates for different users. These dynamic tags reveal user diverse preferences on video contents from different aspects. For instance, the tags of *Michelin*, *Yummy food* and *New York* unearth different concentrations on the video in Fig. 1. When a user clicks a tag, he/she will enter the corresponding *tag channel*, which displays videos only related to the clicked tag. The tag channel can provide an immersive experience for users who need continuous tag-specific video consumption. It contributes nearly 45% video watching time for heavy users in WeChat Top Stories. A better personalized tag list could (1) explicitly show video contents and highlight user's diverse preferences in videos, (2) attract users to click more tags, and (3) guide users to enter the tag channels and watch more tag-specific videos. In conclusion, tag ranking is essential in real-world tag-enhanced video recommendation.

*Both authors contributed equally to this research. Ruobing Xie is the corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3416021

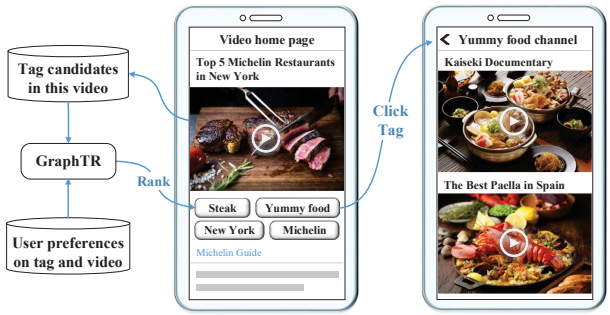

Figure 1: An example of the tag-enhanced video recommendation in WeChat Top Stories. The left screen is the home page which displays videos with personalized tags. The right screen is the tag channel containing videos only related to the specific tag (e.g., yummy food).

In this paper, we concentrate on a novel task, which focuses on *tag ranking in tag-enhanced video recommendation* to improve both tag and video related performances (e.g., tag CTR and video watching time). The main challenges of this task locate in two aspects: (1) tag click behaviors are extremely sparse compared to video click behaviors, which limits supervised model training. (2) This task aims to improve both tag and video related performances, while there are gaps between these two objectives. However, most conventional tag ranking models merely focus on tag-video relevance or tag behaviors, caring less about user preferences on videos [9]. Therefore, we propose a novel **Graph neural network based tag ranking (GraphTR)** in tag-enhanced video recommendation system, which considers heterogeneous interactions between videos, tags, users and medias. Specifically, GraphTR mainly consists of three modules: (1) Heterogeneous network construction, which constructs a huge heterogeneous network to capture global informative interactions between different objectives. These heterogeneous interactions connect related nodes with multi-step paths indicating diverse reasons. (2) Network representation learning, which uses a novel HFIN model with multi-field transformer, GraphSAGE and neural factorization machines (FM) for node aggregations. We also propose an unsupervised learning framework with a neighbor-similarity based loss to learn node representations. And (3) Online tag ranking, which ranks tags according to learned node embeddings and user historical behaviors. GraphTR smartly learns user preferences on tags from multiple interactions (especially from rich video related behaviors and profiles), which alleviates the sparsity issue of tag clicks and bridges the gap between tag/video related objectives.

In experiments, we conduct offline and online evaluations with detailed ablation tests and case studies on a real-world tag-enhanced video recommendation system in *WeChat Top Stories*, which is widely used by millions of people. The significant improvements verify the effectiveness and robustness of GraphTR on both tag and video related metrics. The main contributions of this work are concluded as follows:

- We first highlight the novel task of tag ranking in the tag-enhanced video recommendation system, and propose a new GraphTR framework. To the best of our knowledge, we are the first to bring in graph neural networks for tag ranking in tag-enhanced video recommendation.
- We propose a novel GNN model, which jointly considers multi-field transformer, GraphSAGE and neural FM aggregators with a neighbor-similarity based objective for unsupervised training. It is also the first attempt to combine these aggregators in GNN.
- The significant improvements in both online and offline evaluations verify the effectiveness and robustness of GraphTR on both tag and video related performances. We have deployed GraphTR on *WeChat Top Stories*.

2 RELATED WORK

Conventional tag ranking. Tag ranking and tag recommendation are similar tasks which aim to rank tags [16] or recommend tag sets [23] for given objects. Existing tag ranking methods can be categorized into three classes, namely content-based methods, behavior-based methods and hybrid methods [1]. In content-based methods, NLP tools like Topic models [7] and sequence models [11, 22] are usually used to extract semantic textual features in item contents. [9] focuses on saliency detection to find essential components from visual features. For behavior-based methods, [20] uses tensor factorization for personalized tag recommendation. [30] improves the performances with an attention mechanism. Neural ranking models such as FM [19], DeepFM [8] and AutoInt [24] are also useful. Since user preferences usually change rapidly in practical, Wang et al. [27] further consider temporal effects. Conventional tag ranking task mainly aims to provide appropriate tags to describe the target items (e.g., image or video), which concentrates more on the tag-item relevance. In contrast, our tag ranking task focuses on user preferences on both tags and videos. GraphTR aims to attract users to (1) click more tags, and (2) watch more videos. Conventional tag recommendation is more like a pre-processing module to select relevant tags as candidates for our tag ranking.

Recommendation System. Recommendation system is essential for users to get information [3]. Factorization machine (FM) [19] is a classical method for recommendation that models second-order feature interactions. DeepFM [8] and AutoInt [24] are enhanced with deep neural networks to model high-order feature interactions. Recently, Graph neural network has also been successfully used in recommendation systems. Wu et al. [29] builds its item graph according to sessions. Fan et al. [6] further uses GNN in social recommendation with user and item information. In this paper, we explore the novel problem of how dynamic tag ranks influence the performances in video recommendation. Differing from conventional recommendation tasks, our tag ranking task does not change the video ranks in recommendation system.

Graph neural networks (GNNs). Recently, Graph neural networks have been widely verified in network representation learning (NRL) [5]. Deepwalk [18] conducts random walk on graphs to learn representations. Graph convolution network (GCN) [15] introduces convolution to GNN for classification, and GraphSAGE [10] improves GCN to avoid operating on the entire graph Laplacian. GAT [26] brings attention into GraphSAGE when conducting node aggregation. HetGNN [31] and HAN [28] are designed for heterogeneous networks, considering different types in aggregation. [21] and [13]

further introduce self-attention to GNN. In GraphTR, we implement a novel GNN model, which combines Transformer, GraphSAGE and FM layers for node aggregation with different feature fields. To the best of our knowledge, we are the first to bring in GNN for tag ranking in tag-enhanced video recommendation.

3 METHODOLOGY

GraphTR aims to give a personalized tag list for each user-video pair. The ultimate goal of GraphTR is to improve both tag-related (e.g., tag CTR) and video-related performances (e.g., video watching time) in video recommendation.

3.1 Overall Architecture

The GraphTR framework mainly consists of three modules, namely heterogeneous network construction, network representation learning, and online tag ranking. First, we build a heterogeneous network to capture interactions between videos, tags, users and medias. The multi-hop paths link heterogeneous nodes that are similar in user preference. Next, we use a novel GNN model with the collaboration of multi-field transformer, GraphSAGE and FM aggregators to learn node representations under a neighbor-similarity based unsupervised learning objective. In online tag ranking, we build user preference embeddings from user historical behaviors, and rank all tag candidates according to the similarities between user preference embeddings and tag embeddings.

3.2 Heterogeneous Network Construction

We focus on four different types of heterogeneous nodes including *video*, *tag*, *media* and *user*. Video is the central object in video recommendation and each video belongs to a media (i.e., video provider like BBC). Tags contain both concrete entities and arbitrary concepts, which reflect user diverse preferences from different aspects and granularities. For user nodes, to alleviate data sparsity in user behaviors and accelerate model training, we cluster individual users into user groups with their gender-age-location attribute triplets.

To alleviate the tag sparsity issue, we *make full use of the rich video related behaviors and profiles instead of the sparse tag behaviors* to connect heterogeneous nodes in the network. We attempt to transfer the user preference on videos to tags for tag ranking. Precisely, we select five different types of heterogeneous interactions between these four nodes as our edges. *Video-video* edges are the most dominating interactions based on video watching sequences. To denoise low-quality watching behaviors, we only use the **valid watching behaviors**, where videos are watched for more than 30% of their total time lengths. We generate video-video edges between two videos if they appear next to each other in a valid watching behavior sequence (i.e., video session) of any user. *Video-user* edges are built if a video is validly watched by a user group more than 3 times in a week. *Video-tag* edges connect videos with their tag candidates, while *video-media* edges are drawn between videos and their corresponding providers. We also build *tag-tag* edges when two tags appear in the same videos. All edges are undirected with no weights for convenience. In this case, similar tags are connected via similar user groups, medias and video sessions. These heterogeneous paths provide different recommendation reasons from various aspects. Table 1 gives the detailed statistics of this network, and Sec. 5.6 gives ablation tests on node types.

3.3 Network Representation Learning

Network representation learning aims to learn aggregated node representations of all nodes. We propose a new GNN model **Heterogeneous field interaction network (HFIN)**, which jointly uses transformer, GraphSAGE and neural FM aggregators. Fig. 2 gives the 2-layer architecture of HFIN.

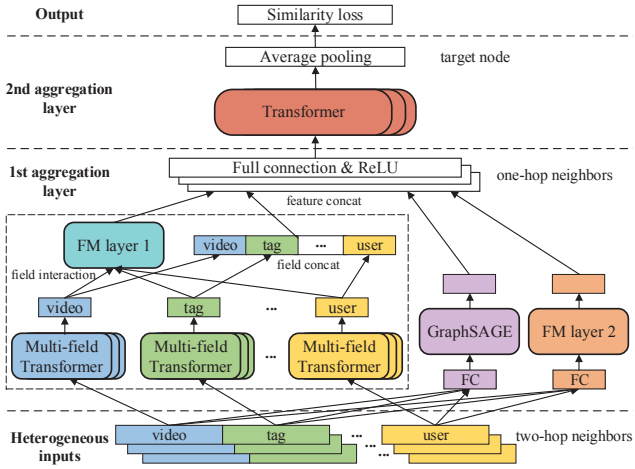

Figure 2: The overall architecture of HFIN.

3.3.1 Heterogeneous Feature Layer. We first project all heterogeneous nodes into the same feature space. For the k-th node and its neighbors N_k, we can divide N_k into four *fields* according to their neighbors' types as $\{\bar{\mathbf{v}}_k, \bar{\mathbf{t}}_k, \bar{\mathbf{m}}_k, \bar{\mathbf{u}}_k\}$. $\bar{\mathbf{v}}_k$, $\bar{\mathbf{t}}_k$, $\bar{\mathbf{m}}_k$ and $\bar{\mathbf{u}}_k$ indicate the summed one-hot representations of video, tag, media and user neighbor sets respectively. The node feature embedding $\mathbf{f}_k$ is concatenated as follows:

$$\mathbf{f}_k = \text{concat}(\hat{\mathbf{v}}_k, \hat{\mathbf{t}}_k, \hat{\mathbf{m}}_k, \hat{\mathbf{u}}_k), \tag{1}$$

where $\hat{\mathbf{v}}_k$ represents the video-field feature embedding. We have $\hat{\mathbf{v}}_k = \mathbf{P}_v \bar{\mathbf{v}}_k$, where $\mathbf{P}_v \in \mathbb{R}^{d_v \times n_v}$ is the lookup projection matrix from $\bar{\mathbf{v}}_k$ to the feature space. n_v is the number of video nodes and d_v is the dimension of $\hat{\mathbf{v}}_k$. For efficiency, the projection matrix is pre-defined as an indicator of the top-frequent video neighbors and fixed during training. Other field feature embeddings $\hat{\mathbf{t}}_k$, $\hat{\mathbf{m}}_k$ and $\hat{\mathbf{u}}_k$ are similar as $\hat{\mathbf{v}}_k$.

3.3.2 Multi-field Interaction Layer. This layer is the first aggregation layer of HFIN, which aggregates two-hop neighbors of the target node to form the one-hop neighbor embeddings. To better capture interactions between different neighbors and fields, this layer consists of three aggregators including multi-field transformer, GraphSAGE and FM aggregators.

Multi-field transformer aggregator. We take the one-hop neighbor's embedding $\mathbf{f}_s$ and its neighbors' node feature embeddings $\{\mathbf{f}_1, \cdots, \mathbf{f}_n\}$ as inputs and consider different fields separately. We conduct multi-head self-attention over neighbors following

Vaswani et al. [25]. Taking the video field as instance, the query, key and value are generated from the video-field feature matrix $\mathbf{F}_v = \{\hat{\mathbf{v}}_s, \hat{\mathbf{v}}_1, \cdots, \hat{\mathbf{v}}_n\}$ as:

$$\mathbf{Q} = \mathbf{W}^Q \mathbf{F}_v, \quad \mathbf{K} = \mathbf{W}^K \mathbf{F}_v, \quad \mathbf{V} = \mathbf{W}^V \mathbf{F}_v, \tag{2}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_h \times d_v}$ are the projection matrices for video field, and d_h is the dimension of queries, keys and values. The self attention is then conducted as:

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

To jointly extract information from different latent subspaces, we also conduct multi-head self-attention as follows:

$$\mathbf{H}_v = \text{concat}(\text{head}_1, \cdots, \text{head}_h) \cdot \mathbf{W}^O, \tag{4}$$

in which the j-th head is calculated as:

$$\text{head}_j = \text{Attention}(\mathbf{W}_j^Q \mathbf{Q}, \mathbf{W}_j^K \mathbf{K}, \mathbf{W}_j^V \mathbf{V}), \tag{5}$$

where $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V \in \mathbb{R}^{d'_h \times d_h}$. We have $d'_h = d_h/h$ with h indicating the number of heads in multi-head attention. $\mathbf{W}^O \in \mathbb{R}^{hd'_h \times d_h}$ is the weighting matrix. Next, we add an average pooling layer to aggregate all $n+1$ output node representations of multi-head transformer as follows:

$$\hat{\mathbf{h}}_v = \text{Average\_pooling}(\mathbf{H}_v), \quad \hat{\mathbf{h}}_v \in \mathbb{R}^{d_h}. \tag{6}$$

$\hat{\mathbf{h}}_t, \hat{\mathbf{h}}_m, \hat{\mathbf{h}}_u$ of other fields are similar as $\hat{\mathbf{h}}_v$. We further conduct a neural FM layer inspired by [12] to capture field-level second-order interactions *after self-attention* as:

$$\mathbf{h}_{\text{FM1}} = \sum_{i=1}^{4} \sum_{j=i+1}^{4} \hat{\mathbf{h}}_i \odot \hat{\mathbf{h}}_j, \quad \hat{\mathbf{h}} = \{\hat{\mathbf{h}}_v, \hat{\mathbf{h}}_t, \hat{\mathbf{h}}_m, \hat{\mathbf{h}}_u\}. \tag{7}$$

$\mathbf{h}_{\text{FM1}} \in \mathbb{R}^{d_h}$ represents the output of FM layer. $\odot$ denotes the element-wise product. Finally, the output of multi-field transformer aggregator $\mathbf{h}_{\text{Trans}}$ is defined as:

$$\mathbf{h}_{\text{Trans}} = \text{concat}(\mathbf{h}_{\text{FM1}}, \hat{\mathbf{h}}_v, \hat{\mathbf{h}}_t, \hat{\mathbf{h}}_m, \hat{\mathbf{h}}_u). \tag{8}$$

The multi-head transformer captures neighbor interactions in each field separately for node aggregation, while the FM layer captures high-level field interactions after transformer.

GraphSAGE aggregator. We use the GraphSAGE aggregator to conduct neighbor aggregation with linear transformation and activation on the whole features [10]. We have:

$$\mathbf{h}_{\text{Graph}} = \text{ReLU}(\mathbf{W}^G \cdot (\sum_{i=1}^{n} \mathbf{f}_i + \mathbf{f}_s)). \tag{9}$$

$\mathbf{W}^G$ is the projection matrix. The input is the feature embedding combination of two-hop neighbors $\mathbf{f}_i$ and the one-hop neighbors $\mathbf{f}_s$ itself. We use ReLU as the activation [17].

FM aggregator. We also conduct a neural FM aggregator that captures second-order interactions between *raw features* of different fields. For video field as example, we first calculate the aggregated field embedding $\mathbf{h}'_v$ over its field feature matrix $\mathbf{F}_v$ by average pooling across neighbors as:

$$\mathbf{h}'_v = \text{Average\_pooling}(\mathbf{F}_v) \cdot \mathbf{W}_v, \quad \mathbf{h}'_v \in \mathbb{R}^{d_h}. \tag{10}$$

$\mathbf{W}_v \in \mathbb{R}^{d_o \times d_h}$ helps to project different field embeddings into the same space. Next, we implement a neural FM as:

$$\mathbf{h}_{\text{FM2}} = \sum_{i=1}^{4} \sum_{j=i+1}^{4} \mathbf{h}'_i \odot \mathbf{h}'_j, \mathbf{h}' = \{\mathbf{h}'_v, \mathbf{h}'_t, \mathbf{h}'_m, \mathbf{h}'_u\}. \tag{11}$$

$\mathbf{h}_{\text{FM2}}$ is the output embedding. Finally, we concatenate the outputs of three aggregators to generate the final node hidden representation of the multi-field interaction layer:

$$\mathbf{h} = \text{concat}(\mathbf{h}_{\text{Trans}}, \mathbf{h}_{\text{Graph}}, \mathbf{h}_{\text{FM2}}). \tag{12}$$

Different parts capture different types of feature interactions.

3.3.3 The Second Aggregation Layer. In the second aggregation layer of HFIN, the input matrix $\mathbf{H} = \{\mathbf{h}_t, \mathbf{h}_1, \cdots, \mathbf{h}_m\}$ is the combination of the target node $\mathbf{h}_t$ and its m one-hop neighbors. It is first fed into a full connection layer. Next, we conduct a classical transformer with an average pooling layer over $m+1$ nodes to get the final aggregated representation $\mathbf{o}$ for all types of target node as:

$$\mathbf{o} = \text{Average\_pooling}(\text{Transformer}(\text{ReLU}(\mathbf{W}^F \mathbf{H}))). \tag{13}$$

$\mathbf{W}^F$ is the weighting matrix of the full connection layer. It is also not difficult to add new fields in HFIN.

3.4 Neighbor-similarity Based Objective

GraphTR focuses on both tag and video related performances, while the natural sparsity of tag clicks limits the supervised learning (see Sec. 5.1 for details). Hence, we attempt to use the rich information of *video-related behaviors and profiles* to learn *user preferences on tags*. Therefore, we creatively design a novel unsupervised learning framework with a neighbor-similarity based loss. It assumes that different types of nodes should be similar to their neighbors.

Precisely in Eq. (13), all types of aggregated representations $\mathbf{o}$ are viewed being projected to the same user preference vector space, where users, videos, tags and medias are connected according to the heterogeneous network. Through the neighbor-similarity based loss, two nodes could be learned similarly via the heterogeneous multi-hop paths that connect them. For example, two videos that are watched by similar users, appear in a video session, or share the same tags/media will be learned similarly. In this case, all heterogeneous node representations (including tag representations) are encoded with different user preferences that mainly derive from rich video related behaviors and profiles, which solves the tag click sparsity issue. The neighbor-similarity based objective can be regarded as a specialized DeepWalk [18] with the path length set as 2 (too long paths may bring in noises), which is formalized as:

$$J = -\sum_{o_i} \sum_{o_k \in N_i} \sum_{o_j \notin N_i} (\log(\sigma(\mathbf{o}_i^\top \mathbf{o}_k)) + \log(1 - \sigma(\mathbf{o}_i^\top \mathbf{o}_j))). \tag{14}$$

N_i is the neighbor set of o_i, and $\sigma(\cdot)$ is the sigmoid function. This loss is a cross entropy loss with neighbor pairs considered as positive samples. We use Adam [14] with negative sampling for training. The main advantage of the neighbor-similarity based objective is that: it makes full use of rich video related behaviors and profiles to encode user preferences on videos into tag representations, which connects isolated nodes in a joint user preference vector space. The neighbor-similarity based loss can also be easily adapted to other GNN-based models if the supervised information is insufficient.

4 ONLINE DEPLOYMENT

In this section, we will give a detailed introduction on the online tag ranking module and our online system and serving.

4.1 Online Tag Ranking

Online tag ranking aims to rank tag candidates for each video and user. To reduce sparsity, we use user's video watching behaviors to extract user preferences on tags and generate user representations. For a video watching sequence $\{v_1, \cdots, v_k\}$, we first calculate the weighting score of the i-th tag as:

$$\alpha_i = \sum_{j=1}^{k} x(ij) \times complete_j \times time_j. \tag{15}$$

$x(ij)$ equals 1 only if the i-th tag t_i is one of the tag candidates in v_j, while $complete_j$ represents the watching time length percentage of v_j. Videos with higher $complete_j$ deserve higher weights. We also bring in $time_j$ to highlight the short-term user interests. We have:

$$time_j = \eta \cdot time_{j+1}, \quad time_k = 1. \tag{16}$$

$\eta = 0.95$ is a time decay factor. Next, we put the tags with top 10 weighting scores into the user tag set T_u, and build user representation $\mathbf{u}$ with these weighted tags as follows:

$$\mathbf{u} = \sum_{t_i \in T_u} \beta_i \mathbf{t}_i, \quad \beta_i = \frac{\alpha_i}{\sum_{t_i \in T_u} \alpha_i}. \tag{17}$$

$\mathbf{u}$ is aggregated by weighted final aggregated tag embeddings $\mathbf{t}_i$ in Eq. (13). We do not use user group embeddings learned by HFIN as user representations, since they are just rough representations of user groups, while user historical behaviors are more informative. Finally, we directly rank all tag candidates of the target video according to the cosine similarities between tag and user preference embeddings. All node representations are learned in offline, stored in a dictionary and fixed in online. Therefore, the time complexity of online tag ranking is $O(\log(n_t k))$ (n_t is the average tag number of video), which is extremely fast.

4.2 Online System and Serving

We have deployed GraphTR on a well-known tag-enhanced video recommendation in *WeChat Top Stories*. *WeChat* is the most popular instant messaging APP in China, which has nearly 1.2 billion active users per month. *WeChat Top Stories* is an integrated recommendation (including video recommendation) application in WeChat ecosystem, which has billion-level daily views. GraphTR is deployed in the video home page, which is triggered after the matching and ranking processes when videos have been recommended. Video/tag lookup tables are needed for tag ranking. In online, we will display top 2 tags from the perspective of user experience and user interface. As shown in Fig. 1, if a user clicks a tag, he/she will enter the corresponding tag channel for immersive video watching related to the clicked tag. We have deployed GraphTR for more than 6 months, affecting millions of users per day.

5 EXPERIMENTS

In experiments, we conduct offline and online evaluations on a real-world tag-enhanced video recommendation system in *WeChat Top Stories* with both tag and video related metrics.

5.1 Datasets

Since there is no large-scale tag ranking dataset that contains both tag and video click behaviors, we build a new dataset WeChat-TR from *WeChat Top Stories* for tag ranking in tag-enhanced video recommendation. WeChat-TR collects nearly 8.6 billion user behaviors of 12 million randomly selected users. We use the tag/video click behaviors in the first few days as train set, and the 231 thousand tag click behaviors in the rest few days as test set for tag CTR prediction. For GraphTR, we build a huge heterogeneous network with nearly 1.8 million nodes and 0.4 billion edges following Sec. 3.2, where users are clustered into 84 thousand user groups. Most videos have multiple tag candidates and the average tag number of videos is 4.96. All tag candidates are annotated manually by editors with the precision above 99% to ensure the tag-video relevance. All datasets have been preprocessed via data masking to protect users' privacy. We do not use existing tag ranking datasets such as Flickr [23], since they do not have behaviors on the tagged objects (e.g., clicks on video) and the tasks are different. The detailed statistics of WeChat-TR are listed in Table 1.

Table 1: Statistics of the WeChat-TR dataset.

video	tag	user	media	tag click	video click
1.5M	113K	12M	93K	341K	8.6B

video-video	video-tag	video-user	video-media	tag-tag
374M	7.4M	3.6M	1.5M	1.5M

5.2 Competitors and Our Methods

Since there is no work specifically designed for this task, we implement some competitive tag ranking models as baselines.

Classical Methods. We first conduct a straightforward Random model that provides random tag ranks for all videos. It considers the diversity of tags in recommendation. Moreover, we implement a Popularity-based model that ranks tags with the popularity of tags learned from user clicks in the overall system.

Content-based Methods. In this paper, we implement two typical content-based methods TF-IDF [2] and BERT [4] as baselines. TF-IDF ranks tag candidates according to their TF-IDF scores in video titles. BERT uses the semantic similarities between tag embeddings and aggregated video title embeddings in user historical behaviors for ranking. All embeddings are pre-trained by BERT and fine-tuned on video titles and the dimension of word embedding is 256. These methods only consider semantic information.

Behavior-based Methods. We implement three powerful ranking models FM [19], DeepFM [8] and AutoInt [24] as our behavior-based methods. We find that simply using tag clicks can only get unsatisfactory results, since tag click behaviors are extremely sparse compared to video click behaviors as shown in Table 1. To solve this, we assume that if a user clicks a video, all tags in this video are viewed as being clicked by this user. In this case, we generate nearly 36 billion "implicit" tag clicks from 8.6 billion video clicks for supervised training with a CTR-oriented objective. All user and item embeddings share the same dimension of 128.

Our GraphTR Methods. We implement three classical models as different node aggregators for ablation tests. The compared models include the enhanced heterogeneous version of DeepWalk [18], GraphSAGE [10] and HGAT [26] (similar as [28]). In experiments, we use GraphTR (X) to represent different GraphTR versions with X as the node aggregator.

5.3 Experimental Settings

In GraphTR, all versions share the same dimension of input features d_f as 750, in which the video-field feature dimension d_v is 300 and other dimensions are 150. The hidden state dimension of final node embedding and d_h is also 150. The node sampling numbers are 30 and 20 for the first and second layers. In online tag ranking, we consider top 200 most recent watched videos in user behaviors. In training, we conduct 20 negative samples for each node pair, with the batch size to be 512. We conduct a grid search for parameter selection. In online system, we display top-2 tags for each video. All models follow the same experimental settings in evaluation.

5.4 Offline Tag CTR Prediction

In offline, we evaluate GraphTR with baselines on WeChat-TR for tag CTR prediction.

5.4.1 Evaluation Protocol. We conduct all models to generate tag ranks for all instances in test set. We regard the hit rate (HIT@N) and mean reciprocal rank (MRR) as our evaluation metrics. In HIT@N, if the clicked tag is ranked in top N, this instance will be regarded as correct. Since our system shows top-2 tags for each video, we use HIT@2 for evaluation.

Table 2: Results in offline tag click prediction.

Model	HIT@2	MRR
Random-based	0.420	0.210
Popularity-based	0.467	0.237
TF-IDF [2]	0.456	0.228
BERT [4]	0.532	0.293
FM [19]	0.476	0.242
DeepFM [8]	0.492	0.259
AutoInt [24]	0.511	0.275
GraphTR (DeepWalk)	0.593	0.330
GraphTR (GraphSAGE)	0.613	0.346
GraphTR (HGAT)	0.623	0.354
GraphTR (HFIN)	**0.678**	**0.384**

5.4.2 Experimental Results. Table 2 demonstrates the evaluation results, from which we can observe that:

(1) All GraphTR models significantly outperform all baselines on HIT@2 and MRR with the significance level $\alpha = 0.01$. It confirms that GraphTR can generate better dynamic tags that are more related to the video contents and user interests, and thus could attract users to click more tags. The deviation of HFIN is ±0.003 for HIT@2 and ±0.002 for MRR. Differing from most conventional tag ranking models, we use unsupervised structural information on heterogeneous interactions instead of supervised but rare tag click information. It verifies that the neighbor-similarity based loss can handle cold-start scenarios. We also split the test set according to the number of tag candidates, and find that the improvements are more significant when videos have more tag candidates.

(2) The content-based methods TF-IDF and BERT perform better than Random but worse than GraphTR, which implies that the semantic information contributes less to our tag ranking task. TF-IDF tends to recommend low-frequent tags, while these unique tags may not be welcomed by users in video recommendation. In contrast, BERT tends to recommend tags that are semantically similar to the videos located in user historical behaviors, while the semantic similarity may not always lead to the similarity in user preference. Differing from conventional tag ranking models that mainly focus on tag-content similarity, our task concentrates more on user preferences in tags. The gap between semantic similarity and user preference similarity limits the content-based tag ranking methods in our tag ranking scenario.

(3) The behavior-based methods like FM, DeepFM and AutoInt achieve better results compared to TF-IDF but are still worse than GraphTR. We consider the implicit tag clicks generated from video clicks as supervised information, which inevitably brings in noises. We also use the original sparse tag clicks to train behavior-based models, while the results are even worse due to the insufficient training. These results indicate that classical behavior-based tag ranking methods cannot work well in our scenarios where tag behaviors are sparse. It also implies that the proposed neighbor-similarity based objective is essential for training GraphTR.

(4) Comparing different GraphTR models, we find that HFIN achieves the best performances on both metrics. It confirms the power of three multi-field aggregators. Specifically, multi-field transformer considers neighbor-level interactions separately in each field, GraphSAGE focuses on neighbor aggregation with node features as a whole, and FM models the field-level feature interactions of raw features. All aggregators are essential for node aggregation in HFIN. In Sec. 5.6, we further conduct several ablation tests on different HFIN components and different types of nodes.

5.5 Online Tag/Video Related Evaluation

The improvements in offline evaluation verify the effectiveness of GraphTR on tag click prediction, while our tag ranking task also aims to improve video-related performances, which are hard to be evaluated in offline. Hence, we further conduct an online A/B test with both tag-related and video-related metrics.

5.5.1 Online System and Evaluation Protocol. We evaluate GraphTR on a tag-enhanced video recommendation system named WeChat Top Stories as shown in Fig. 1, which is used by millions of users. All videos in the home page are attached with some tags generated by tag ranking models. After clicking a tag, users will enter the tag channel that only contains videos related to the clicked tag.

We implement five tag ranking models to compare with the Random baseline, and focus on both home page and tag channel scenarios with the following evaluation metrics: (1) Tag click rate. (2) Tag click number per capita. (3) Tag list-wise click rate. (4) Average video watching time. (5) Video views per capita. (6) Page turns per capita. The former three metrics measure the tag-related performances in the home page. In contrast, the latter three metrics

Table 3: Online A/B test on tag-related metrics and video-related metrics in WeChat Top Stories.

	Tag click rate	Tag click number	Tag list-wise click rate	Average video watching time	Video views per capita	Page turns per capita
BERT	+4.64%	+2.27%	+3.47%	+4.49%	+1.46%	+1.89%
GraphTR (DeepWalk)	+5.46%	+2.92%	+3.92%	+3.15%	+1.77%	+2.89%
GraphTR (GraphSAGE)	+5.74%	+3.53%	+4.11%	+6.25%	+4.31%	+4.49%
GraphTR (HGAT)	+7.30%	+4.58%	+5.60%	+7.23%	+4.73%	+4.96%
GraphTR (HFIN)	**+8.48%**	**+5.60%**	**+6.81%**	**+9.70%**	**+5.11%**	**+6.32%**

measure the implicit impacts of recommended tags on the video-related performances in tag channels. We conduct the online A/B test for 3 weeks, with nearly 50 million people influenced by our tag ranking models. We report the improvement percentages instead of specific values. The online evaluation could be viewed as an online ablation test for different NRL models in GraphTR.

5.5.2 Experimental Results. Table 3 shows the results on tag-related and video-related metrics. We observe that:

(1) All GraphTR models significantly outperform all baselines, among which GraphTR (HFIN) achieves the best performances on all evaluation metrics. The significance level of the improvements brought by HFIN is $\alpha = 0.01$. We have also passed the A/A homogeneity test in online evaluation, which confirms that the system and traffic split are unbiased and the deviation of two same models is not significant. It verifies the effectiveness and robustness of our models in real-world scenarios.

(2) The improvements in tag-related metrics reconfirm that GraphTR can recommend appropriate tags that attract users to click. Tag click rate is a classical CTR metric that measures user satisfaction on tags, while Tag list-wise click rate focuses more on the whole tag list. Tag click number per capita implies the influence of tags on users. All these metrics confirm the advantages of GraphTR on tag-related performances from different aspects.

(3) The improvements in video-related metrics indicate that GraphTR could generate better personalized tags, which even benefits the core video-related indicators (e.g., video views and watching time). This is astonishing since we *do not even change the video ranks*. The better tags we recommend, the more users are willing to enter the tag channels that they truly like, and the more time they will spend on watching tag-specific videos. The page-turning behaviors reflect user satisfaction in the tag channel, for it implies that users are interested and willing to see more videos in tag channels.

(4) HFIN outperforms all NRL models in node aggregation of GraphTR. It is because that the self-attention model could make full use of the informative interactions between different nodes. Moreover, the neural FM layers also successfully capture field-level interactions, which bridge the gaps between heterogeneous neighbors and fields. Detailed ablation tests are in Sec. 5.6.

5.5.3 Improvements on the Home Page. We further conduct an A/B test for video related performances in the home page. We observe that all GraphTR models have slightly better or comparable performances compared to baselines. GraphTR (HFIN) achieves 0.72% improvements in video CTR with the significance level $\alpha = 0.01$. Moreover, it achieves 0.43% improvements in the percentage of valid watching behaviors (see Sec. 3.2). These astonishing results indicate that GraphTR could even benefit video performances in the home page without changing video ranks. Currently, tag-related behaviors are still sparse compared to those of videos. The impacts of tag ranking models will be much more significant if users are more involved with tags and tag channels.

5.6 Ablation Tests

We further conduct an ablation test to show the effectiveness of different components and nodes in HFIN. Table 4 shows the results of different GraphTR versions. We find that:

(1) All components including the 1st/2nd multi-field transformer, GraphSAGE and FM layers are indispensable in HFIN. Precisely, the 1st transformer is the main source of feature interactions for field-specific neighbor aggregation, and thus HFIN has a significant decline without the 1st transformer. The GraphSAGE and FM work as supplements to provide whole-feature-level and field-level interactions of raw features, which also bring in significant improvements. The 2nd transformer enables multi-step aggregation to build the target node representations. We have tried to add a third transformer layer while the performance is only comparable.

(2) We also evaluate the importance of different types of nodes and interactions in Heterogeneous network construction. Since videos and tags are the basic objects in tag-enhanced video recommendation, we wipe out user nodes and media nodes with their edges as ablation tests. We find that both user and media nodes are significant in GraphTR, since they provide additional information to connect similar nodes with different types of user preferences (e.g., similar tags may be connected with multi-step paths via video sessions, similar medias and related users).

Table 4: Ablation tests for GraphTR.

Ablation version	HIT@2	MRR
GraphTR (HFIN)	**0.678**	**0.384**
– GraphSAGE layer	0.671	0.380
– FM layer	0.662	0.371
– 1st transformer layer	0.617	0.341
– 2nd transformer layer	0.597	0.332
– user nodes and edges	0.643	0.359
– media nodes and edges	0.656	0.367

5.7 Case Study

5.7.1 Tag Embeddings. Table 5 shows some nearest tags calculated by cosine similarities with tag embeddings in GraphTR (HFIN).

For instance, users that have watched videos related to *The Great Wall* may also be interested in *Monument* or other ancient Chinese buildings like *Forbidden city*. Besides, users interested in history or tour are also willing to seek information of *Qin Dynasty* (when *The Great Wall* was built) or *Tourism inventory*. These nearest tags reflect not only semantic similarities on tags, but also user preferences and videos. We also conduct a quantitative analysis on 100 randomly-sampled top-frequent tags with human annotators, which shows that the percentage of diversified tag (the tag that has at least 3 tags belonging to different categories in top 5 nearest tags) is 89%.

Table 5: Examples of target tags with nearest tags.

Tag	Nearest tags
The Great Wall	Monument; World Cultural Heritage; Forbidden city; Qin Dynasty; Tourism inventory
Michelin star restaurant	Michelin chef; French red wine; Gourmet show; Spain Seafood Risotto; Japanese food
New energy vehicle	Hydrogen powered vehicle; Fuel consumption; Baojun; 4-wheel drive; Foreign car

5.7.2 Personalized Tag Ranking. Table 6 gives a real dynamic tag case for different users. User1 is a fan of N Jia (an actor) and loves variety shows (e.g., Go Champion!). User2 is crazy about basketball and its superstars like Jordan and O'Neal. User3 is simply interested in funny videos with no preferences in specific actors or stars. GraphTR well captures these user preferences and explicitly shows different personalized tags to highlights different contents. Hence, all users are attracted and willing to click tags and watch this video.

Table 6: Tag ranking results for different users.

Video title	**Shaquille O'Neal** performs his Dream shake and **N Jia** imitates the movement **comically**.
User1 tags	**N Jia**; **Go Champion!**; Variety show in China
User2 tags	**Shaquille O'Neal**; **Basketball**; Variety show
User3 tags	**Imitation**; Funny moment; Variety show

6 CONCLUSION AND FUTURE WORK

In this paper, we highlight the tag ranking in tag-enhanced video recommendation. We propose a novel GraphTR, which creatively uses a new HFIN model to combine transformer, GraphSAGE and FM for node aggregation on heterogeneous networks. GraphTR utilizes rich information in video-related behaviors and profiles to learn user preferences on tags. Both online and offline evaluations confirm the significant improvements in tag and video related metrics. GraphTR has been deployed on a real-world tag-enhanced video recommendation system in WeChat Top Stories.

In future, more interactions like social relations and tag-related behaviors could be considered in network construction. Weighted edges could also be used in our network. Moreover, we will design more sophisticated NRL models and online ranking models with supervised learning to improve the performances, and enhance the user nodes with more sophisticated representations.

REFERENCES

[1] Fabiano M Belém, Jussara M Almeida, and Marcos A Gonçalves. 2017. A survey on tag recommendation methods. *JASIST* (2017).
[2] Iván Cantador, Alejandro Bellogín, and David Vallet. 2010. Content-based recommendation in social tagging systems. In *Proceedings of RecSys*.
[3] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of RecSys*.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.
[5] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. In *Proceedings of KDD*.
[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *Proceedings of WWW*.
[7] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of WWW*.
[8] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of IJCAI*.
[9] Jingfan Guo, Tongwei Ren, Lei Huang, and Jia Bei. 2019. Saliency detection on sampled images for tag ranking. *Multimedia Systems* (2019).
[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of NIPS*.
[11] Hebatallah A Mohamed Hassan, Giuseppe Sansonetti, Fabio Gasparetti, and Alessandro Micarelli. 2018. Semantic-based tag recommendation in scientific bookmarking systems. In *Proceedings of RecSys*.
[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of SIGIR*.
[13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *Proceedings of WWW*.
[14] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.
[15] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.
[16] Dong Liu, Xian-Sheng Hua, Linjun Yang, Meng Wang, and Hong-Jiang Zhang. 2009. Tag ranking. In *Proceedings of WWW*.
[17] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*.
[18] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of KDD*.
[19] Steffen Rendle. 2010. Factorization machines. In *Proceedings of ICDM*.
[20] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of WSDM*.
[21] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2019. Dynamic Graph Representation Learning via Self-Attention Networks. In *Proceedings of CIKM*.
[22] Xuewen Shi, Heyan Huang, Shuyang Zhao, Ping Jian, and Yi-Kun Tang. 2019. Tag Recommendation by Word-Level Tag Sequence Modeling. In *Proceedings of DASFAA*.
[23] Börkur Sigurbjörnsson and Roelof Van Zwol. 2008. Flickr tag recommendation based on collective knowledge. In *Proceedings of WWW*.
[24] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of CIKM*.
[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*.
[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of ICLR*.
[27] Keqiang Wang, Yuanyuan Jin, Haofen Wang, Hongwei Peng, and Xiaoling Wang. 2018. Personalized time-aware tag recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[28] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *Proceedings of WWW*.
[29] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *Proceedings of AAAI*.
[30] Jiahao Yuan, Yuanyuan Jin, Wenyan Liu, and Xiaoling Wang. 2019. Attention-Based Neural Tag Recommendation. In *Proceedings of DASFAA*.
[31] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of KDD*.